

Treball de Fi de Màster

Màster Universitari en Enginyeria Industrial

Estimació de la producció d'energia solar/consum emprant xarxes neuronals

MEMÒRIA

Autor: Fermin Lumbreras Ederra
Director: Ramon Costa Castelló
Convocatòria: Setembre de 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



RESUM:

Per a l'elaboració d'aquest treball de Fi de Màster, es van dissenyar xarxes neuronals artificials amb l'objectiu de fer previsions de la generació i consum energètic d'una micro xarxa alimentada per panells fotovoltaics. Per dissenyar les xarxes neuronals artificials, es van utilitzar històrics de dades de generació i consum d'aquesta micro xarxa. El software utilitzat és *Matlab*, més concretament, les seves eines de *Deep Learning* i disseny de xarxes neuronals artificials.

ABSTRACT:

In order to elaborate this Master's Degree Final Thesis, artificial neural networks were made, with the aim of make predictions about the energetic generation and consumption of a micro grid powered by photovoltaic panels. To design the artificial neural networks, historic data set of generation and consumption of this micro grid was used. The employed software is *Matlab*, more specifically, its tools of *Deep Learning* and artificial neural network design.

LABURPENA:

Master Amaierako Lan honetarako, sare neuronal artifizialak egin dira, panel fotovoltaikoekin ekoitzitako mikro-sare baten energia generazio eta kontsumoaren inguruko iragarpenak betetzeko helburuarekin. Sare neuronal artifizialak diseinatzeko, mikro-sare honen generazio eta kontsumoen historikoak erabili ziren. Erabilitako softwarea *Matlab* da, zehazki, bere *Deep Learning* eta sare neuronal artifizialak diseinatzeko tresnak.

Paraules clau:

Xarxa neuronal artificial

Deep Learning

Predicció

Matlab

Energies renovables

Sumari

SUMARI	9
1. PREFACI	11
1.1. Origen del projecte	11
1.2. Motivació.....	11
1.3. Requeriments previs.....	11
2. INTRODUCCIÓ	13
3. ESTAT DE L'ART	15
3.1. Introducció.....	15
3.1.1. Neurones: estructura biològica.....	15
3.1.2. Adaptació matemàtica	16
3.2. Xarxa neuronal artificial clàssica	18
3.2.1. Composició d'una xarxa neuronal artificial clàssica	18
3.2.2. Funcionament de la xarxa neuronal artificial clàssica	22
3.3. Xarxa neuronal artificial recurrent	25
3.3.1. Definició.....	25
3.3.2. Xarxa <i>LSTM</i>	25
3.4. Problemes a l'entrenament de la xarxa neuronal artificial	27
3.4.1. Valors inicials dels paràmetres	27
3.4.2. Tassa d'aprenentatge	27
3.4.3. <i>Overfitting</i>	28
3.4.4. Escalament dels dades d'entrada	28
3.4.5. Arquitectura de la xarxa	28
3.4.6. Mínims múltiples.....	28
4. METODOLOGIA DEL PROJECTE	29
4.1. Dades utilitzades	29
4.1.1. Introducció	29
4.1.2. Dades de generació de la base de dades australiana	29
4.1.3. Dades de generació de la microxarxa del projecte.....	31

4.1.4.	Dades del consum de la microxarxa del projecte	33
4.2.	Organització de les dades.....	34
4.3.	Software: <i>Matlab</i>	36
5.	RESULTATS	41
5.1.	Predicció generació australiana	41
5.2.	Predicció generació dades del projecte	42
5.3.	Predicció consum dades del projecte.....	48
6.	CONCLUSIONS I MIRES CAP AL FUTUR	53
6.1.	Conclusions sobre les prediccions fetes de la generació	53
6.2.	Conclusions sobre les prediccions fetes del consum	54
6.3.	Mires cap al futur	54
BIBLIOGRAFIA		57

1. Prefaci

1.1. Origen del projecte

Aquest projecte va ser penjat a la intranet de TFM de l'ETSEIB. El departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI) va llançar la proposta, amb l'objectiu de dur a terme aquest projecte en col·laboració d'una tesi doctoral.

1.2. Motivació

La meua missió, un cop acabats els meus estudis del Màster Universitari d'Enginyeria Industrial, és d'emplenar les meves esforços i energia en el desenvolupament d'energies renovables. Per això, amb aquest projecte, vaig trobar un enllaç perfecte entre el fi de la meua etapa universitària i el començ d'un nou camí al món laboral, dins de les energies renovables.

1.3. Requeriments previs

Per un costat, per a l'execució d'aquest projecte, van ser necessaris uns coneixements bàsics del software informàtic *Matlab*. Durant els estudis de grau, vaig desenvolupar en prou signatures projectes amb *Matlab*, pel que em va ser més fàcil comprendre les tasques previs.

Per altre costat, va ser requerit una comprensió mínima sobre l'energia fotovoltaica. Aquest projecte no només tracta d'un desenvolupament d'una programació informàtica; és necessari comprendre i viure el fenomen fotovoltaic.

2. Introducció

La predicció de la producció i consum energètic té una gran rellevància per als sistemes i tecnologies que tenen una gran aleatorietat, així com la energia solar fotovoltaica. Les energies renovables han de guanyar importància al model actual d'implementació d'energia, però com s'ha dit abans, tenen una component altíssima de incontrolabilitat.

Per altre costat, hi ha una part de la ciència informàtica que està experimentant, des de finals del segle XX, una evolució exponencial: l'aprenentatge automàtic. Amb aquesta eina, es pot simular i fer prediccions d'esdeveniments aleatoris, que no obeeixen a comportaments lineals, tal com la radiació solar que fa que funcionin els generadors fotovoltaics.

En aquest treball es pretén desenvolupar uns algorismes que siguin capaços de predir la producció energètica d'una micro xarxa alimentada amb panells fotovoltaics, i altres algorismes que també prediguin el consum energètic d'aquesta micro xarxa. En ambdós casos s'haurà d'emplenar mecanismes de *Deep Learning*. Per alimentar les xarxes s'utilitzaran històrics de dades.

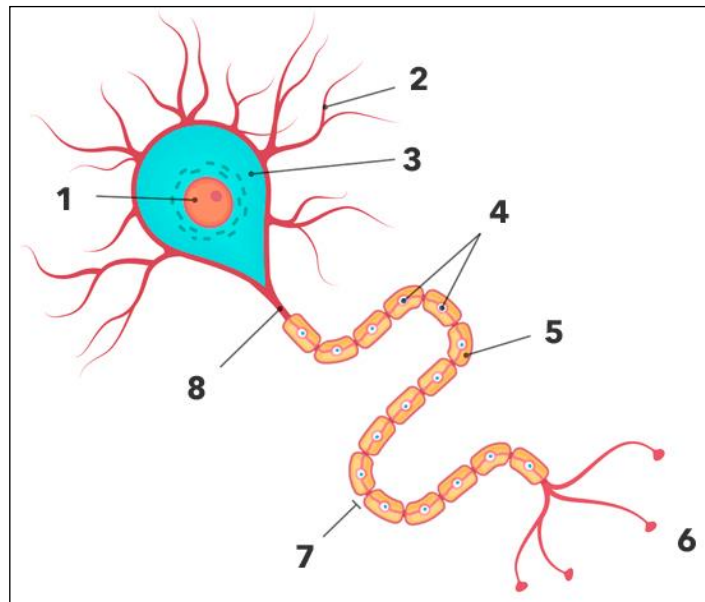
El software utilitzat per aquest projecte és *Matlab*, el qual disposa d'eines interessants per al desenvolupament de xarxes neuronals artificials, així com d'ajudes i components per a la comprensió d'aquest programa informàtic.

3. Estat de l'art

3.1. Introducció

3.1.1. Neurones: estructura biològica

Les xarxes neuronals artificials estan basats en l'estructura del sistema nerviós humà, on l'element fonamental és la neurona.



Il·lustració 3-3-1: Esquema general d'una neurona humana [1]

Tal com apareix a la *Il·lustració 3-1*, el *nucli* (1) es troba dins del *cos* o *soma* de la cèl·lula (3). D'aquest cos surten unes ramificacions anomenades *dendrites* (2), sent una fibra la més llarga de totes, l'*axó* (8). L'*axó* es ramifica en *filaments* i *subfilaments*, a di de connectar-se amb les dendrites i cossos de les cèl·lules d'altres neurones; a aquesta connexió se li coneix com sinapsis. Dins de qualsevol cervell humà, cadascuna de les neurones pot tenir entre dotzenes i centenars de milers de connexions amb altres neurones.

El rest d'elements que componen la neurona són la *cèl·lula de Schwann* (4), la *mielina* (5) i el *Node de Ranvier* (7); com es veurà més endavant, endinsar-se en explicacions sobre aquests

elements no té gaire importància per aquest projecte.

Les senyals que es transmeten d'una neurona a altre, ocorren mitjançant una reacció electroquímica. Les sinapsis alliberen substàncies químiques transmissores i entren a la dendrita, provocant una elevació en el potencial elèctric del cos de la cèl·lula. Un cop superat el límit establert, s'envia a l'axó un impuls elèctric; això es difondre mitjançant les rames de l'axó, arribant a les sinapsis i donant transmissors en los cossos de les neurones receptores. Les sinapsis que pugen el potencial s'anomenen *excitadores*, i les que baixen el potencial, *inhibidores*.

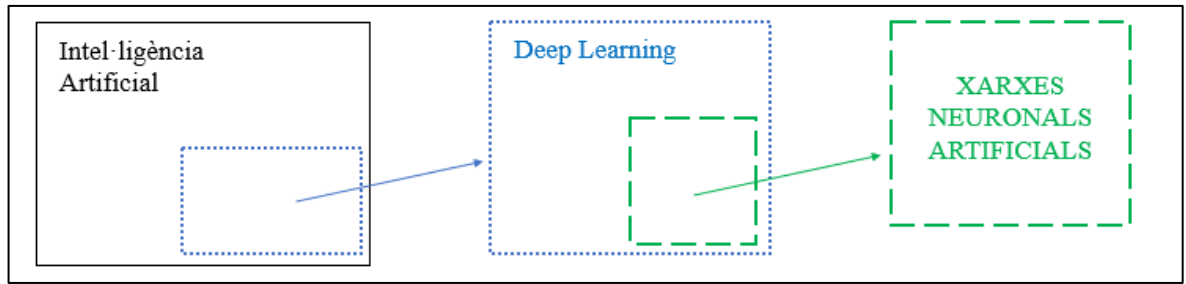
Aquestes connexions sinàptiques posseeixen una plasticitat clau per fer canvis, a llarg termini, de les intensitats de les connexions com resposta a un patró de simulació. Es a dir, tenen la capacitat d'actualitzar-se al llarg del temps. Això constitueix el fonament de l'aprenentatge dins del cervell humà.

3.1.2. Adaptació matemàtica

El fonament de la intel·ligència artificial va ser el modelatge del comportament d'un cervell humà. Des de la part final de la primera meitat del segle XX, es van desenvolupar models matemàtics que van tenir analogies amb el comportament de les neurones humans, com abans s'ha descrit [2].

Aquests algorismes van sorgir per resoldre problemes que tenen una quantitat de dades elevada i no es poden aplicar fàcilment les regles convencionals d'anàlisi matemàtic.

Breument, com es mostra a la *Il·lustració 3-3-2*, d'una banda, les xarxes neuronals estan dins de les tecnologies de aprenentatge profund o *Deep Learning* i d'altra banda, el *Deep Learning* forma part de una de les moltíssimes que conformen la intel·ligència artificial.

*Il·lustració 3-2: Esquema principal*

Una xarxa neuronal artificial (a partir d'ara s'anomenarà ANN, atès al seu nom en anglès: *Artificial Neural Network*) és una tècnica de *Machine Learning*, més concretament, de *Deep Learning* o *Aprenentatge Profund*, que envolten molt mètodes matemàtiques. Les ANN posseeixen la capacitat de resoldre un ampli rang de problemes, ja que el seu comportament matemàtic s'assoleix al de una composició de neurones humans.

Explicat d'una manera ràpida, el principi de funcionament de les ANN és el següent: mitjançant el lliurament de uns dades coneguts a la ANN, aquesta té la capacitat de donar uns dades o informació de sortida, a força dels seus algoritmes i computació interns. En aquest projecte, la sortida desitjada és una predicció dels dades d'entrada.

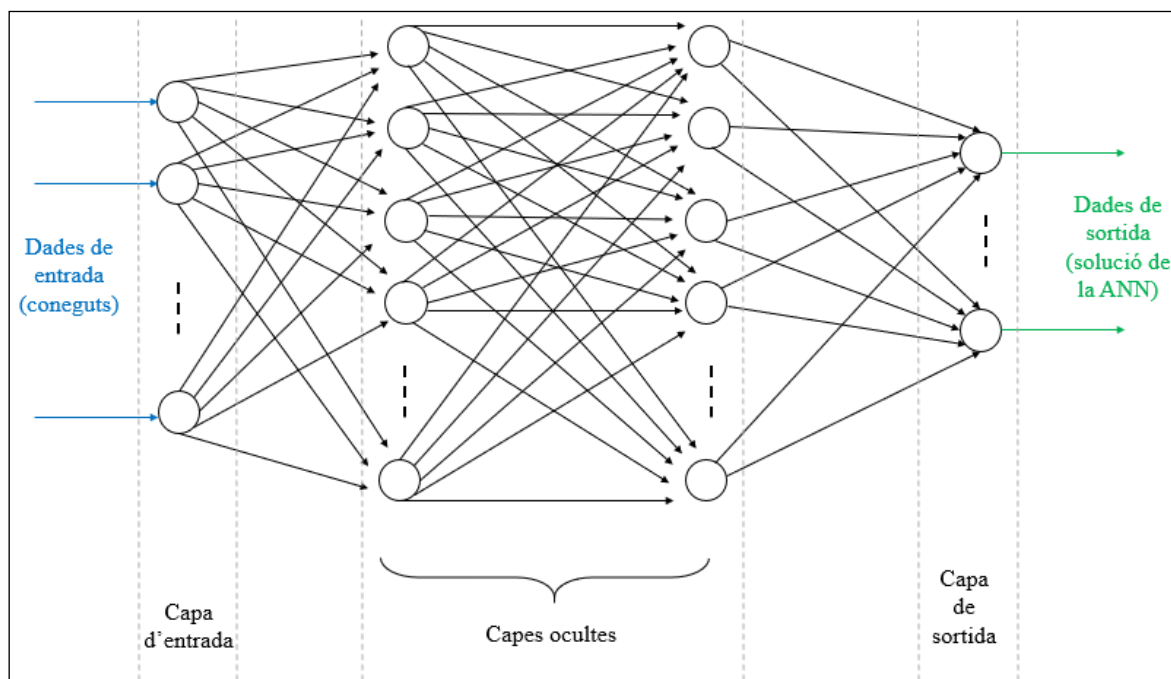
Perquè treballi la xarxa, cal dividir els dades coneguts en 2 grups. D'una banda, amb el primer grup (entre el 70% i 80% dels dades), la xarxa *entrena* (***train***). Així, la xarxa fa les seves prediccions, ajustant els seus paràmetres per a millorar la seva precisió. D'altra banda, amb la resta de les dades, la xarxa *proba* (***test***), també fa comparacions dels seus resultats de predicció amb les dades conegudes, però sense fer canvis a la seva estructura. Així mateix s'estudia si l'arquitectura de la xarxa, per a un problema en particular que tingui que veure amb els dades utilitzat per fer el ***train*** i ***test***, serveix.

3.2. Xarxa neuronal artificial clàssica

3.2.1. Composició d'una xarxa neuronal artificial clàssica

Les capes

Una ANN convencional està disposada com es mostra a la *Il·lustració 3-3*. Està formada per diferents capes, on es troben les neurones. La peculiaritat que hi ha és que les neurones d'una capa estan connectades amb totes les neurones de la seva darrera capa i amb totes les neurones de la següent capa [3].



Il·lustració 3-3: Disposició d'una xarxa neuronal artificial (ANN) convencional

A continuació, s'expliquen les diferents capes:

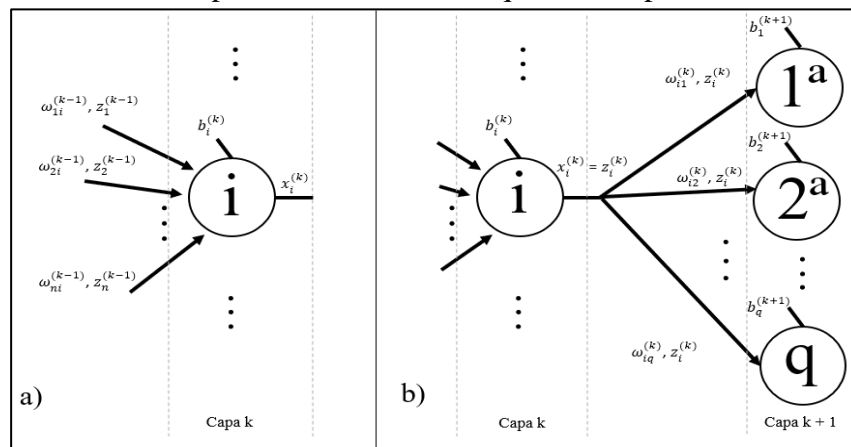
1. **Capa d'entrada (*Input Layer*):** La primera de les capes. El nombre de neurones és el mateix que el de la quantitat de variables d'entrada. A la *Il·lustració 4-4* no està definit el nombre de entrades.

2. **Capes ocultes (*Hidden Layers*):** Aquesta capa es la més complexa de totes, atès que el seu disseny és la part més complexa de la xarxa neuronal. No hi ha una regla explícita sobre qual és la millor composició per aquesta capa; hi ha capes que, encara que tinguin diferent dissenys, ambdues donen bon resultat. A la *Il·lustració 3-3* es mostra una configuració de 2 capes ocultes, sense especificar tampoc el nombre de neurones per capa.
3. **Capa de sortida (*Output Layers*):** La darrera de les capes. El nombre de neurones d'aquesta és el mateix que la quantitat de variables de sortida de la xarxa. No es defineix, a la *Il·lustració 3-3*, el nombre exacte de sortides de la xarxa.

La neurona

Llavors, convé explicar en que consisteix una neurona. És la unitat de treball de la xarxa, i el que fa és tenir un numero o ponderació. A més, en el cas de les neurones a la *Hidden Layers*, intervenen més factors.

1. **Neurona a la *Input Layer*:** La seva funció és la d'enviar a totes les neurones de la capa següent el valor de la variable d'entrada. Normalment, hi ha un pre-tractament, per a normalitzar els dades i que estén entre 0 i 1, per facilitar el funcionament de la xarxa.
2. **Neurona a la *Hidden Layer*:** Aquí l'assumpte és diferent. Tal i com es mostra a la *Il·lustració 4-5*, es pot veure els elements que els componen.



Il·lustració 3-4: Exemple d'una neurona qualsevol (i), a la k-èsima Hidden Layer, mostrant els seus components.

a) Perspectiva de quan rep senyals de les neurones de la capa k-1. b) Perspectiva de quan envia senyals a les neurones de la capa k+1, que també és altre Hidden Layer.

Per un costat, tal i com està a la *Il·lustració 3-4 a*), a la neurona i de la *Hidden Layer* k , l'arriben les sortides de totes les neurones de la darrera capa $k-1$, amb valors de $z_n^{(k-1)}$; la z vol dir que es tracta del valor de sortida de una neurona, el sobre-índex $(k-1)$ indica que la senyal arriba de la capa anterior a la k i el sub-índex n diu qual de les neurones de la darrera capa emeses la senyal. A més a més, hi ha un paràmetre clau a la transmissió de senyals entre neurones: els **pesos** $\omega_{ni}^{(k-1)}$. Globalment, es pot definir com la ponderació que té la connexió entre 2 neurones; tal i com s'ha explicat abans, es refereix al sub-índex ni com quina neurona n de la capa $(k-1)$ està connectada amb quina neurona i de la capa (k) .

Després, a la *Il·lustració 3-4 b*), es veu els senyals de sortida de la neurona i a les neurones de la capa $k+1$. Hi ha un detall important: el valor de sortida d'aquesta neurona ($x_i^{(k)}$) és com si es digués que és el valor d'entrada de una neurona de la següent capa de la senyal que arriba d'aquesta neurona ($z_i^{(k)}$).

A més, cadascuna d'aquestes neurones tenen altre paràmetre associat, el $b_i^{(k)}$, anomenat **bias**. Aquest paràmetre funciona com un llindar; el seu comportament s'explicarà més endavant.

- 3. Neurona a la *Output Layer*:** Les senyals que surten de les últimes capes ocultes (*Hidden Layers*), van a aquestes neurones. Aquí participen les senyals que arriben de la darrera capa i el bias d'aquesta neurona. Normalment, hi ha un post-tractament dels dades, per de-normalitzar i tenir els valor de sortida de la xarxa a la magnitud real.

Funció d'activació

Es introdueixen les dades de entrada a les neurones de la *Input Layer*, i des de aquestes neurones, després del pre-tractament, s'estableixen les connexions amb les neurones de la següent capa (la primera capa de la *Hidden Layer*). El valor d'entrada de cadascuna de les neurones d'aquesta capa és la suma entre les senyals provinents de l'anterior capa multiplicat per els seus corresponents pesos.

A més, en cadascuna de les neurones de aquesta capa passa una operació matemàtica, anomenada *funció d'activació*. És un càlcul mitjançant es decideix si aquesta neurona emeses valor de sortida o no, i en el cas de que ho faci, qual serà el seu valor. Primerament, s'haurà de veure si la suma dels valors de les senyals d'entrada son majors que aquest valor del bias; si no es així, la neurona no donarà una ordre d'activació a la neurona, donant-li un valor de sortida nul.

$$x_i^{(k)} = f\left(\sum_{m=1}^n \omega_{mi}^{(k-1)} \cdot z_m^{(k-1)} - b_i^{(k)}\right)$$

Equació 3-1: Valor de sortida de la senyal de la neurona

Segonament, es donarà a terme la funció d'activació, perquè es necessita que es compleixin diversos requeriments. En primer lloc, es vull que la senyal de sortida de la neurona quedi acotada. En segon lloc, aquesta funció ha de tenir una derivada simple; així, s'estalvia en cost computacional.

Hi ha diverses funcions d'activació, però per aquest projecte només es treballa amb 2 funcions d'activació: la **sigmoidal** i la **tangent hiperbòlica**.

Funció d'activació sigmoidal: Transforma el valor resultant de l'*Equació 3-1* en un valor entre 1 i 0, segons la seva magnitud. Concretament, els valors grans tendeixen de manera asimptòtica cap a 1, al contrari dels valors petits, que tendeixen asimptòticament cap al 0.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Equació 3-2: Funció d'activació sigmoidal

Funció d'activació tangent hiperbòlica: Es comporta de la mateixa manera que la sigmoidal, però agrupant els resultats entre 1 i -1 (es a dir, amb aquesta funció d'activació es pot tenir valors de sortida de positius i negatius).

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

Equació 3-3: Funció d'activació tangent hiperbòlica

3.2.2. Funcionament de la xarxa neuronal artificial clàssica

Entrenament i Test de la xarxa

Per al disseny d'una xarxa neuronal artificial, és necessari disposar d'un conjunt de dades d'entrada a la xarxa neuronal amb les seves sortides conegudes. Per aquest projecte, per exemple, es vull dissenyar una xarxa neuronal per a aconseguir predir la radiació que haurà a un lloc concret de Barcelona. Per obtenir una xarxa vàlida, es prenen valors coneguts del darrer any, i amb ells es fa el disseny.

El disseny de la xarxa neuronal artificial consta de dos passos:

1. **Entrenament:** Cal recordar que les xarxes neuronals artificials imiten el comportament d'un cervell humà. Així mateix, ha de començar aprenent i modificar les variables de la xarxa (els bias, els pesos...), per a aconseguir un modelatge precís de la xarxa que es vull tenir. És a dir, en aquesta fase, els dades d'entrada de la xarxa i els dades de sortida són coneguts. Primer, es propaguen els dades des de l'entrada fins la sortida, on es faran les prediccions. Aquesta primera part de l'entrenament s'anomena **feedforward** (propagació de les senyals d'entrada per la xarxa fins a la sortida). Així doncs, a mesura que es fan les prediccions, aquestes es poden comparar amb els dades de sortida reals. D'aquesta forma, un cop anotat la magnitud de l'error (diferència entre valor real i la predicció feta amb la xarxa), es fan canvis a les variables de la xarxa. Aquesta manera de calcular l'error s'anomena **backpropagation**, però s'explicarà després. Així doncs, aquesta etapa també s'anomena **entrenament** o **training**, i es prenen entre el 60% i el 80% dels dades. A més, fer l'entrenament amb el conjunt de dades sencer és una **època**. Amb tot, mai es considera que la xarxa està entrenada després de només una època; s'han de fer moltes èpoques per considerar que la xarxa estigui entrenada.

2. **Test:** Agafant les dades que falten, aquí es fa ho mateix que a la etapa de l'aprenentatge, només que les variables de la xarxa no es canvien. És a dir, es mesura el error de les prediccions fetes amb la xarxa i els valors reals, però només per veure si la xarxa es bona o si cal fer un canvi estructural al disseny.

Backpropagation

Es tracta de l'algoritme utilitzat al procés d'aprenentatge de la xarxa neuronal artificial. Tal i com s'ha dit abans, després de cadascuna de les prediccions es mesura l'error de sortida de la xarxa, entre la predicció i el valor real de sortida. A l'Equació 3-4 apareixen els seus components: y_m són els valors reals de sortida i z_m les prediccions.

$$E = \frac{1}{2} \cdot \sum_{m=1}^q (y_m - z_m)^2$$

Equació 3-4: Error global després d'una predicció per a tota la capa de sortida

Com s'ha vist en els darrers apartats, la xarxa neuronal que es dissenya prèviament, té uns paràmetres que es van canviant a segons l'error obtingut. Això s'aconsegueix minimitzant l'error segons la magnitud o influència dels pesos i bias: s'utilitza el gradient de l'error que actua sobre els pesos i els bias; per tant, són necessàries les derivades parcials de cadascun d'aquests paràmetres. En efecte, fent el càlcul de l'error quadràtic, a més de mesurar l'error que hi ha, facilita el càlcul de la seva posterior derivada.

L'objectiu de la *backpropagation* és anar modificant els valors dels tots els pesos ω i bias b després de cadascuna de les prediccions, perquè l'error sigui menor. Les expressions de la acció que s'ha de dur a terme pels càlculs del gradient de l'error es mostren a continuació, a les Equació 3-5 i Equació 3-6, respectivament.

$$-\nabla \omega_{mi}^{(k)} = -\frac{\partial E}{\partial \omega_{mi}^{(k)}}$$

Equació 3-5: Gradient genèric de l'error dels pesos que s'ha de reduir

$$-\nabla b_i^{(k)} = -\frac{\partial E}{\partial b_i^{(k)}}$$

Equació 3-6: Gradient genèric de l'error dels bias que s'ha de reduir

La variació dels valors dels pesos i bias es realitzen, com s'ha explicat a la darrera secció, després de cada predicció, però només a la etapa de *training*. Es tracta d'un procés iteratiu, i existeixen molts algorismes que fan les variacions sota els seus criteris; ara bé, tots ells compleixen aquesta condició: s'utilitza un altre paràmetre, anomenat **tassa d'aprenentatge** (α), mitjançant es dur a terme la variació dels paràmetres. Llavors, segons l'error que hi ha, s'haurà de reduir o augmentar el valor dels paràmetres de la xarxa.

$$\Delta\omega_{mi}^{(k)} = -\alpha \cdot \frac{\partial E}{\partial \omega_{mi}^{(k)}}$$

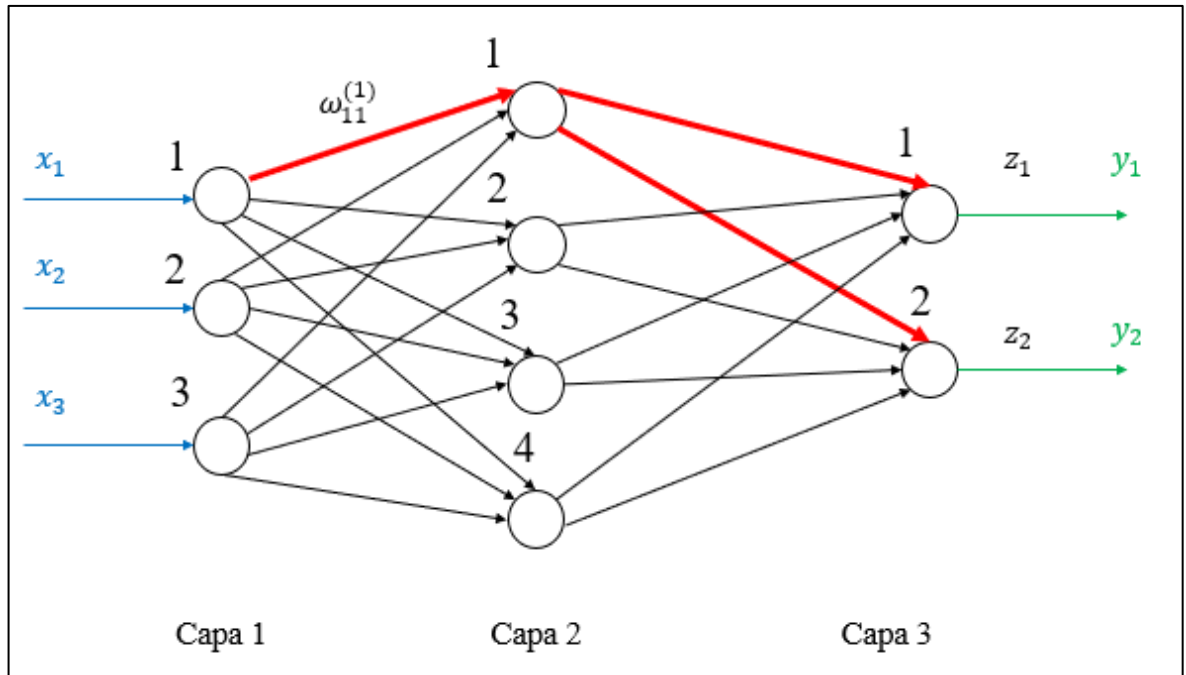
Equació 3-7: Variació del valor dels pesos després de cadascuna de les prediccions

$$\Delta b_i^{(k)} = -\alpha \cdot \frac{\partial E}{\partial b_i^{(k)}}$$

Equació 3-8: Variació del valor dels bias després de cadascuna de les prediccions

El fonament d'aquestes derivades parcials és la regla de la cadena. És a dir, per a cadascun de les paràmetres que es vull fer el reajustament, només es té en compte els “camis” que segueixen fins l'obtenció de l'error. Com mostra esquemàtica, es té la *Il·lustració 3-5*; aquí es veu un cas particular, on es representa una xarxa neuronal artificial bàsica: una capa d'entrada amb 3 neurones, una capa oculta amb quatre neurones i una capa de sortida amb dues neurones.

Així doncs, es pretén analitzar la influència del pes $\omega_{11}^{(1)}$ (el pes que enllaça la neurona 1 de la capa 1 amb la neurona 2 de la capa 2), fins l'error, d'una forma gràfica.



Il·lustració 3-5: Representació esquemàtica de la derivada parcial amb respecte l'error del pes $\omega_{11}^{(1)}$, com exemple de la Regla de la Cadena.

Aquest algoritme permet aconseguir un mínim relatiu; això és atès que, quan major és l'error obtingut, major serà la variació del paràmetre fent la correcció de l'error, i quan l'error sigui petit, la variació serà menys violenta.

3.3. Xarxa neuronal artificial recurrent

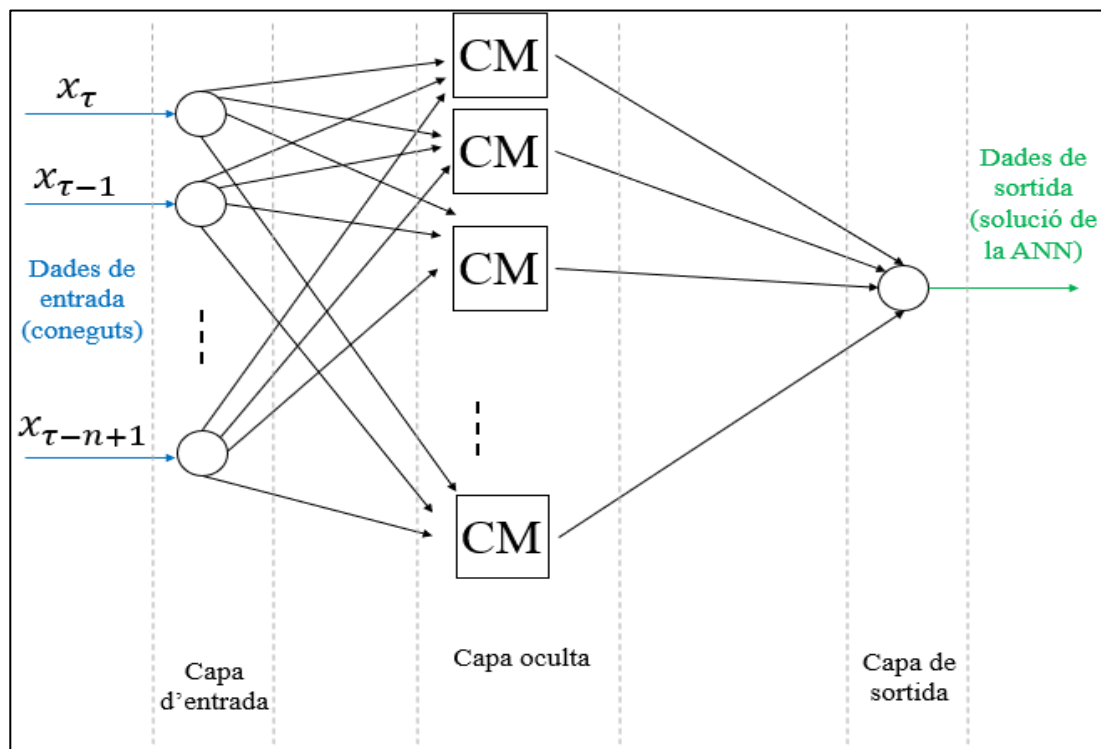
3.3.1. Definició

Es refereix al tipus de xarxa que es dinàmica, es a dir, que té memòria.

3.3.2. Xarxa LSTM

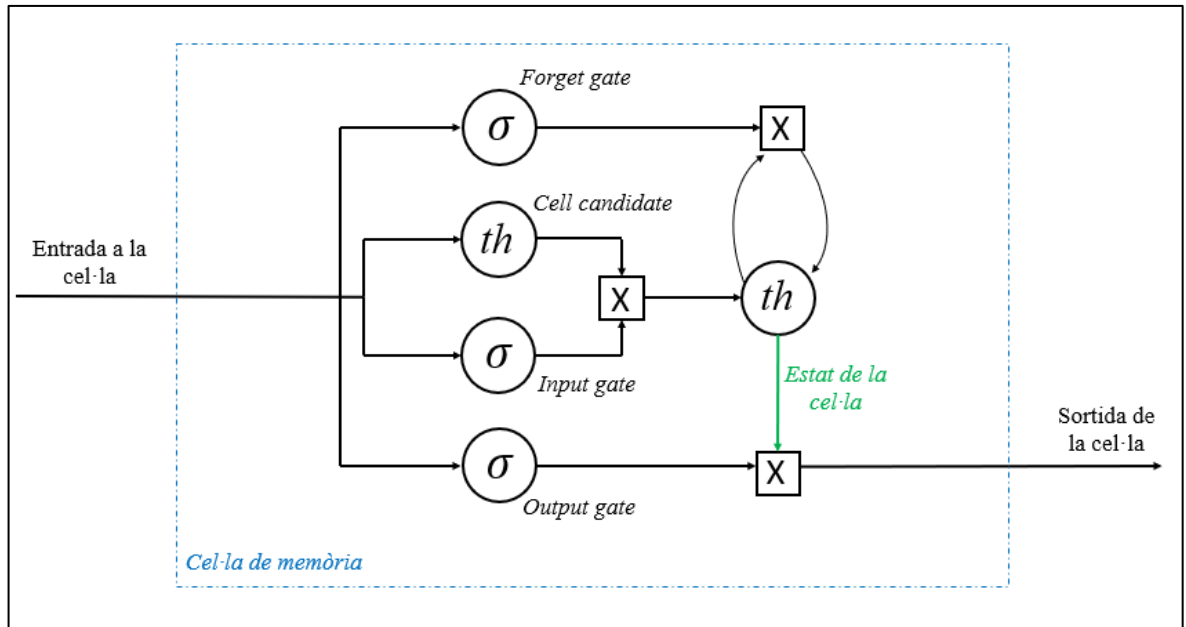
Les xarxes LSTM (de l'anglès, *Long Short-Term Memory*), són uns tipus de xarxes neuronals artificials recurrents que solucionen els problemes que presenten les xarxes neuronals recurrents convencionals [4]. Aquestes xarxes tenen problemes a l'entrenament, atès que els gradients retropropagats són molt irregulars, per culpa de que aquest gradient depèn de l'error actual i dels errors passats. L'acumulació d'errors fa que sigui difícil memoritzar a llarg termini.

Per tant, es diu que les xarxes LSTM son adequades pel tractament de series temporals perquè tenen la capacitat de decidir quina informació ha de ser emmagatzemada, i quina ha de ser eliminada.



Il·lustració 3-6: Xarxa neuronal artificial amb cel·les de memòria LSTM a la capa oculta.

A la Il·lustració 3-6 es mostra una xarxa amb només una capa oculta, formada per cel·les de memòria. A continuació, a la Il·lustració 3-7, es mostra l'interior de aquestes cel·les de memòria.



3.4. Problemes a l'entrenament de la xarxa neuronal artificial

3.4.1. Valors inicials dels paràmetres

Abans d'introduir les dades a la xarxa per a començar amb l'entrenament, s'han d'establir els valors inicials dels pesos i bias de la xarxa. Normalment, es prefereix tenir valors inicials d'aquests paràmetres propers al zero, tenint valors positius i negatius.

3.4.2. Tassa d'aprenentatge

La taxa d'aprenentatge de la *backpropagation* té influència a la solució final. Una taxa massa alta, pot donar resultats insuficients, però una taxa massa petita fa que l'entrenament tingui un cost computacional molt alt. Un bo mètode és escollir un valor per la taxa gran, i anar baixant el seu valor fins tenir una taxa de valor petit amb un temps de computació raonable.

3.4.3. *Overfitting*

Aquest és el major problema que presenten les xarxes neuronals artificials. S'anomena *sobreajust* perquè la xarxa fallarà en el moment de predir resultats, perquè no tenen estrictament els mateixos valors de les mostres de l'entrenament.

S'ha de permetre que la xarxa sigui capaç de tenir les seves conclusions; per això, convé entrenar la xarxa amb valors variats i nombrosos.

3.4.4. Escalament dels dades d'entrada

Es necessari normalitzar els dades d'entrada de la xarxa; és habitual centrar els dades en 0, amb una desviació típica de 1.

3.4.5. Arquitectura de la xarxa

Es prefereix tenir moltes neurones per capa oculta que poques. Amb poques neurones, el model no arribarà a tenir la flexibilitat necessària, i no agafarà l'estructura no lineal de les dades [5].

3.4.6. Mínims múltiples

La solució final del model dependrà del valor inicial dels pesos. Així doncs, és usual començar amb diferents valors, i després escollir el conjunt de valors inicials que donen el menor error.

4. Metodologia del projecte

4.1. Dades utilitzades

4.1.1. Introducció

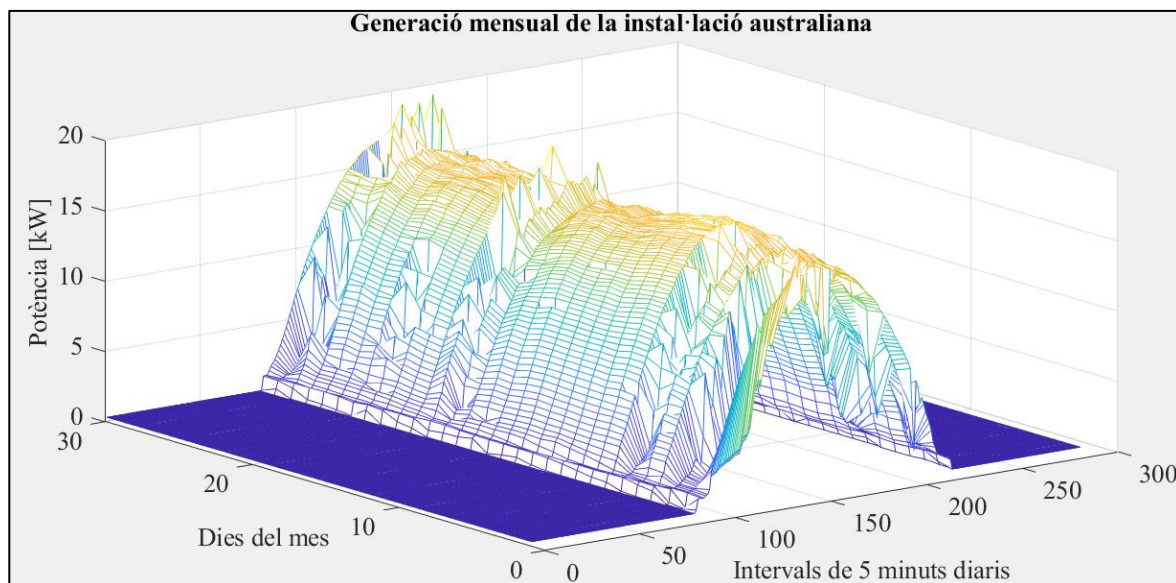
Els dues objectius principals d'aquest projecte són dissenyar una xarxa neuronal artificial vàlida per fer prediccions de la radiació solar a un lloc de Barcelona mitjançant els seus valors passats, i altra xarxa neuronal artificial per veure si, utilitzant les prediccions obtingudes per la xarxa anterior, es poden predir els consums d'aquesta micro xarxa.

A banda d'això, abans de començar a treballar amb aquestes dades, es va utilitzar els dades d'una base de dades d'un camp fotovoltaic australià, que estan penjats a la pàgina web de *DKA Solar Centre* [6]. Això no dona informació que pugui ser utilitzada d'una manera directa a las xarxes neuronals de la micro xarxa del projecte. Amb tot, atès que es tracta d'una sèrie de dades més curta, no es poden generalitzar estratègies per a l'aplicació de les xarxes de les dades d'un any sencer; ara bé, com es veurà a continuació, els dades de generació a la base dades australiana són continus i gairebé no tenen interrupcions, cosa que no passa amb els dades de generació de la micro xarxa. Així doncs, es bo fer unes proves inicials amb les dades australianes i veure com es pot començar a dissenyar una xarxa neuronal artificial.

4.1.2. Dades de generació de la base de dades australià

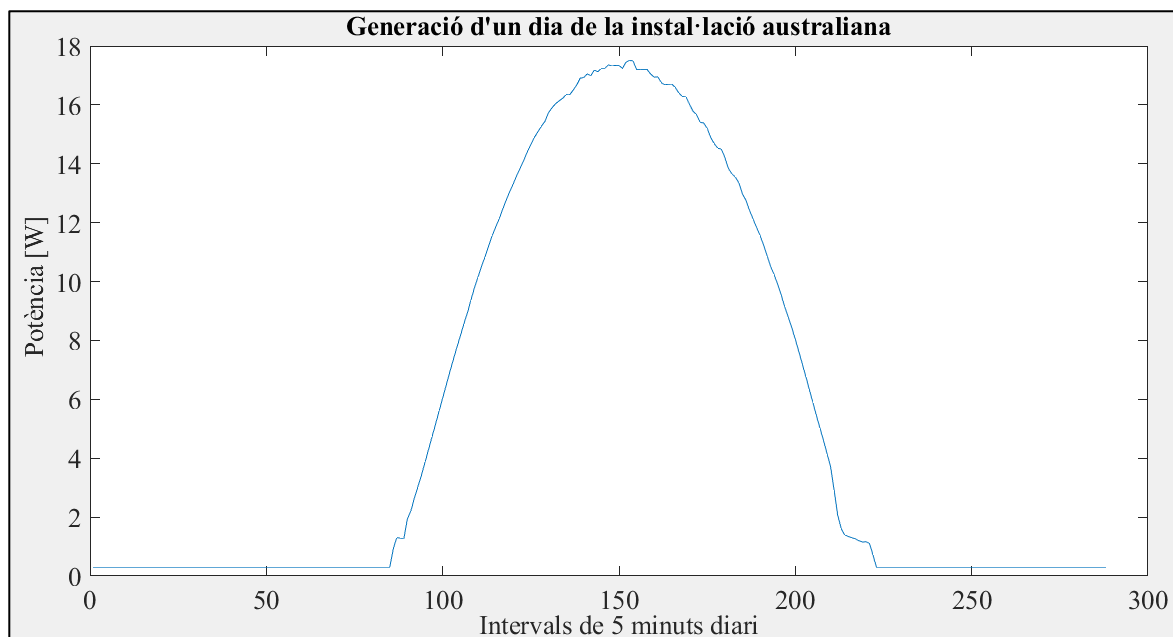
Com s'ha dit abans, atès que l'obtenció de dades és massa difícil, només es pot aconseguir, a la web *DKA Solar Centre*, informació de generació de instal·lacions fotovoltaïques d'un més sencer només (març). Aquestes dades són observacions fetes cada 5 minuts, per la qual cosa es tenen 288 dades de radiació per dia, durant 30 dies.

A la *Il·lustració 4-1* apareixen aquestes dades; es presenten de forma que es pugui veure com la generació és màxima quan la radiació solar és màxima: al migdia. A les hores de nul·la radiació, a les nits, no hi ha cap generació.



Il·lustració 4-1: Distribució diària de la generació d'un mes sencer d'una instal·lació australiana

A la introducció d'aquest capítol es va dir que la radiació gairebé té interrupcions; aleshores, a la Il·lustració 4-2 es representa el novè dia de març, i això es comprova.

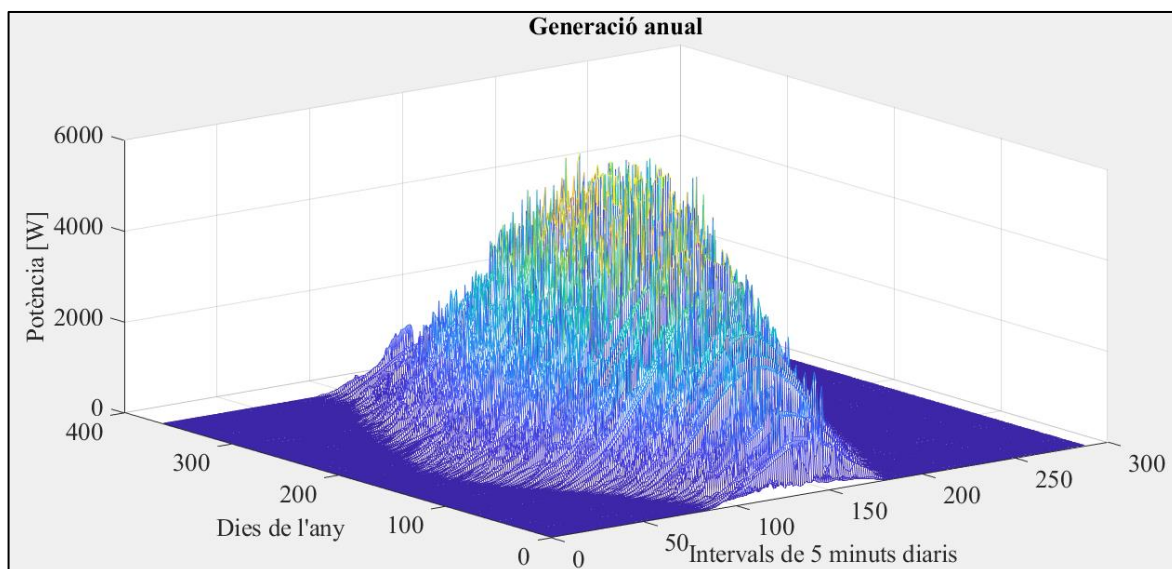


Il·lustració 4-2: Generació d'un dia d'una instal·lació australiana

Llavors, les avantatges d'aquest conjunt de dades podrien ser l'escassa discontinuïtat, a causa

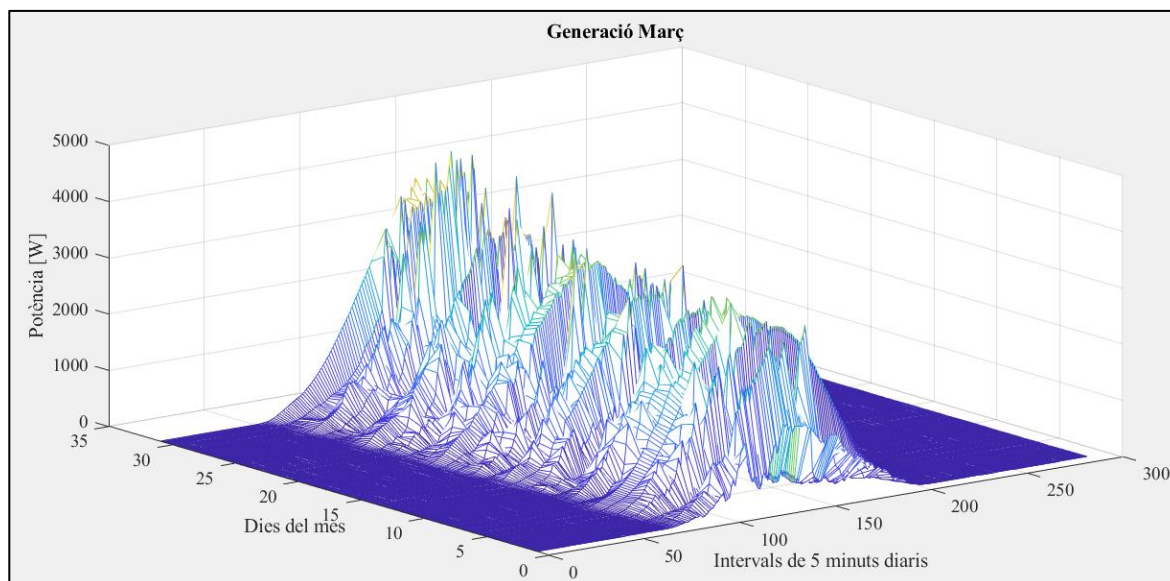
d'una nuvolositat nul·la a Austràlia, i les desavantatges, que no es pot disposar d'un conjunt major d'un mes.

4.1.3. Dades de generació de la microxarxa del projecte



Il·lustració 4-3: Distribució anual de la generació de la micro xarxa del darrer any sencer

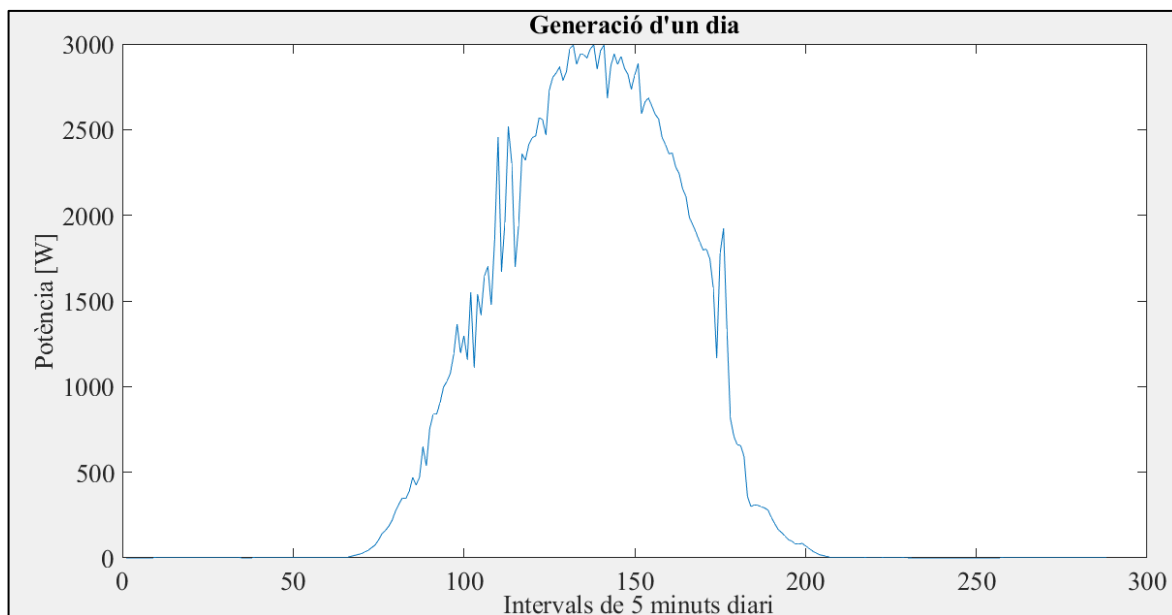
Sobre aquestes paraules, a la *Il·lustració 4-3*, es disposa del conjunt de les dades de generació del any 2018; 288 mostres diaris, per 365 dies. Un cop més, la relació entre la radiació solar i la generació dels mòduls fotovoltaics és directa: a les nits, la generació es nul·la, però a migdia és quan la generació es màxima.



Il·lustració 4-4: Distribució mensual de març de la generació del mes sencer

A més a més, també es nota que en hivern els dies són més curts, i que la radiació solar és menys intensa, mentre que a l'estiu els dies són més llargs i la radiació obtinguda és major.

Per tal de justificar la utilització de les dades australianes, es fan els gràfics de la generació mensual per a març i la generació del novè dia de març, a les *Il·lustració 4-4* i *Il·lustració 4-5*, respectivament. Ambdues imatges mostren que la radiació a la microxarxa situada a Barcelona pateix una variabilitat major que la australiana, ja que la nuvolositat és major. Llavors, el modelatge d'aquestes funcions són menys continus, i per tant, més complicats.

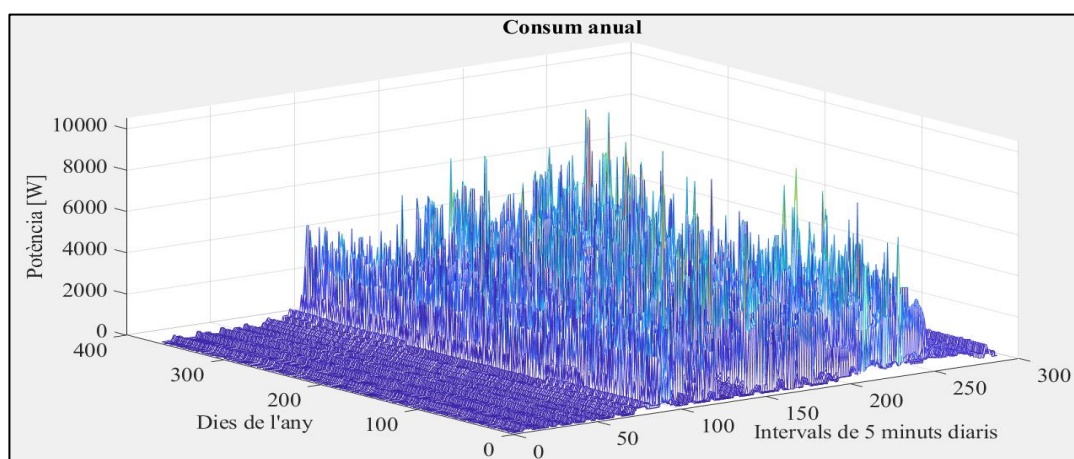


Il·lustració 4-5: Generació d'un dia de març de la micro xarxa del projecte

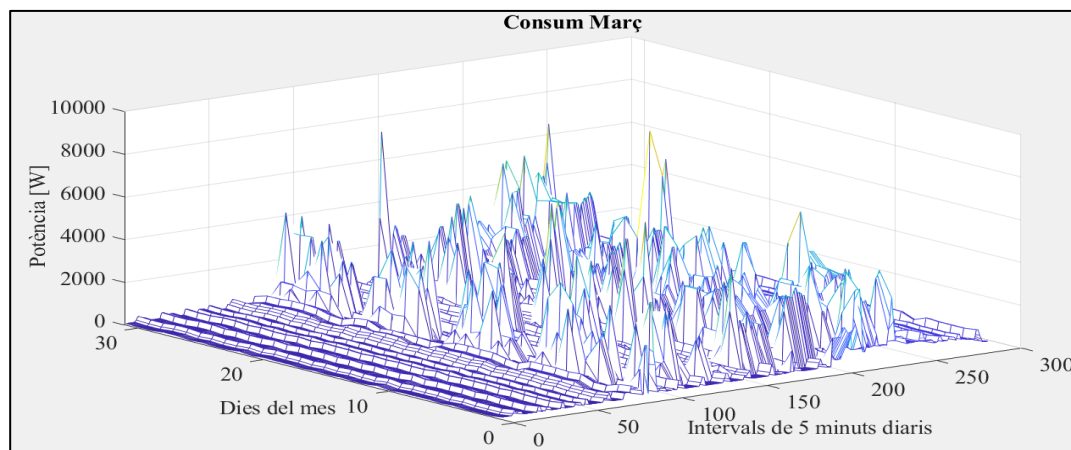
4.1.4. Dades del consum de la microxarxa del projecte

Finalment, es mostren els dades de consum de la microxarxa. Com es veu a la *Il·lustració 4-6*, aquestes dades no tenen variabilitat força gran al llarg del any; només presenten variacions al llarg del dia, tal i com apareix a la *Il·lustració 4-8*.

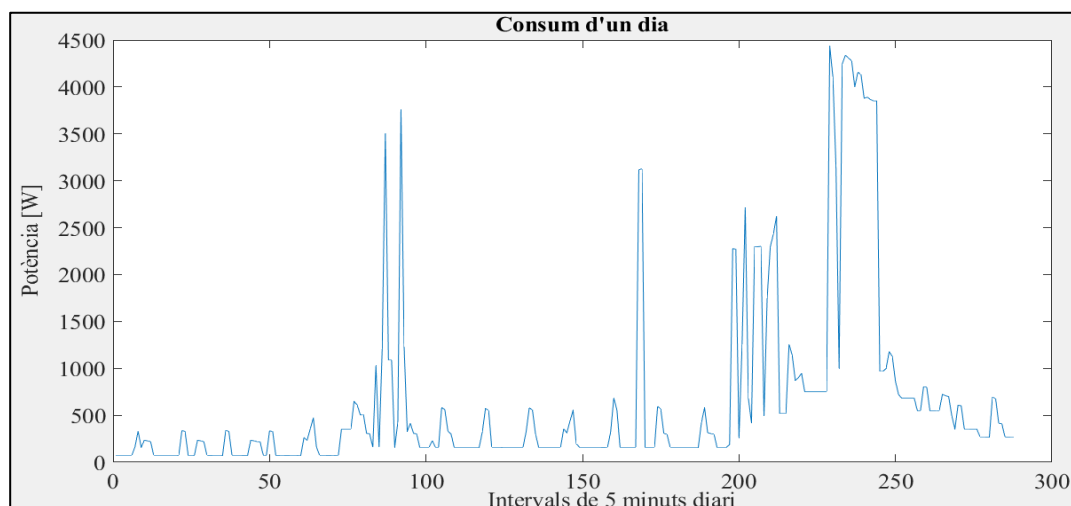
També es presenta, a la *Il·lustració 4-7*, el consum que es va tenir en març del darrer any. Un altre cop, es veu la naturalesa no lineal d'aquesta sèrie temporal.



Il·lustració 4-6: Distribució anual del consum de la micro xarxa del darrer any sencer



Il·lustració 4-7: Distribució mensual de març del consum del mes sencer



Il·lustració 4-8: Consum d'un dia de març de la micro xarxa del projecte

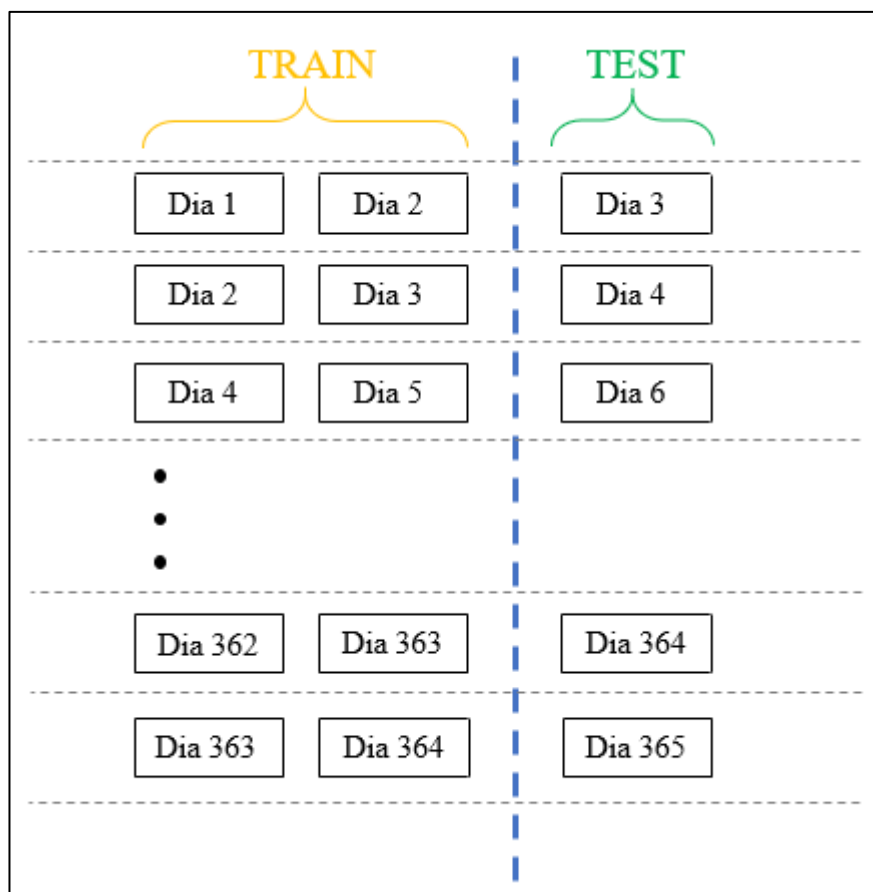
4.2. Organització de les dades

Aquestes mètodes de treball de *Deep Learning*, depenen, amb massa força, de com s'organitzen els dades abans de la feina. Per aquest projecte, es vull treballar amb una xarxa “educada” amb dades coneguts d'un any sencer.

Per això, és fonamental la manera de plantejar el tractament dels dades. Perquè la xarxa pugui tenir coneixements del comportament de la generació que vull predir d'un any sencer, ha de

conèixer aquesta tendència. S'ha fet un entrenament de 2 dies consecutius i al tercer dia, el test; això, aplicat per tots els dies de l'any. És a dir: amb el primer i segon dia de l'any s'entrena per fer el test amb el tercer. A més, es fa altre entrenament amb el segon i tercer dia, per fer el test el quart dia. Es segueix amb aquesta tendència, fins arribar a l'entrenament del 363er i 364rt dia pel test del 365en dia. D'una forma esquemàtica, es mostra aquesta disposició a la *Il·lustració 4-9*. El conjunt de dades resultant, és una matriu d'entrenament de 363 x 576, i una matriu de test de 363 x 288.

Per organitzar els dades de la base dades australiana, només s'ha fet per un mes sencer. Així doncs, la matriu de dades d'entrenament és de 28 x 576, i la de test, 28 x 288.



Il·lustració 4-9: Esquema de la distribució de les dades al codi

A més, per a la introducció de dades a la xarxa, es va normalitzar els dades utilitzats per fer el *train* i el *test*. Agafant el major i menor valor d'aquesta sèrie temporal, es pretén tenir valors d'entrada entre 0 i 1. Així, la xarxa neuronal artificial treballa millor, reduint riscos de que pugui

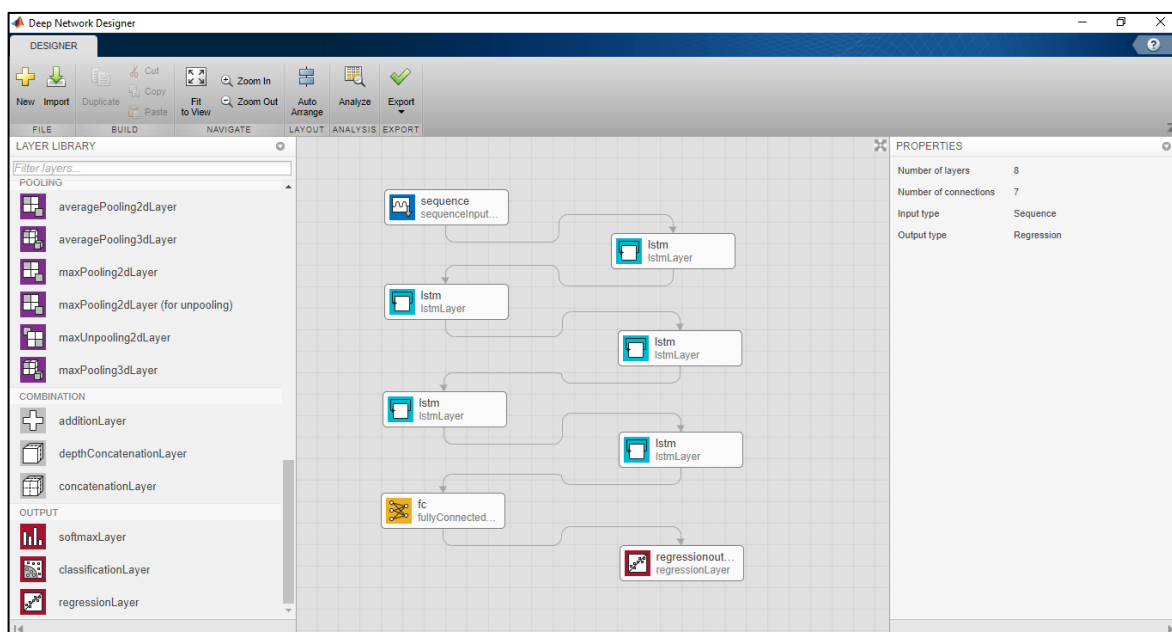
treballar malament.

Una altre decisió presa va ser si es filtraven els dades o no. Per un costat, filtrant els dades fent mitjanes horàries, es podria tenir una reducció considerable de dades i, per tant, utilitzar menys cost operacional; però utilitzant mitjanes només, es perdria una variabilitat de la generació que sí es apareix a les dades cinc-minutals. Per altre costat, hi ha autors que recomanen eliminar les hores nocturnes de generació fotovoltaica [7]; així, també es podria estalviar en costo operacional, però un cop més, es perdria la naturalesa de les dades. A més, si es vol fer una relació entre generació i consum, es necessiten les mateixes dimensions a les dades, i el consum apareix també de nit.

En resum, per a l'elaboració de les xarxes neuronals artificials es treballaran amb dades de dies sencers, amb presa de dades 5-minutals. És a dir, per dia, ha d'haver 288 dades.

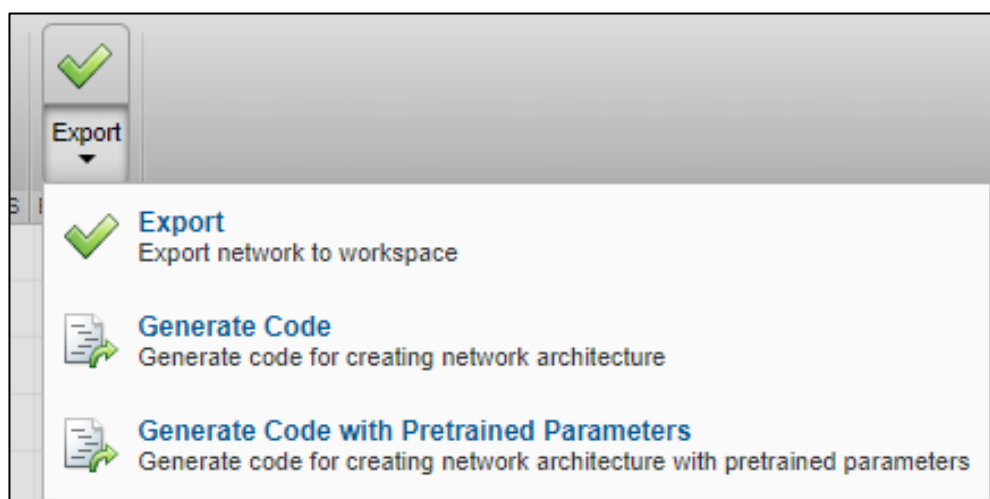
4.3. Software: *Matlab*

En aquest apartat s'han agrupat les tècniques utilitzades per al disseny de les xarxes neuronals artificials, al software informàtic *Matlab*. Primerament, cal dir que aquest software disposa d'un entorn de programació agradable i intuïtiu per al programador. Això és conseqüència de la disponibilitat d'apps que el software té. A la *Il·lustració 4-10* està dibuixada una potencial xarxa neuronal amb l'eina *Deep Network Designer*. Mitjançant aquest programa, es poden realitzar varis tipus de xarxes; convé recordar que aquest treball consisteix en dissenyar xarxes neuronals per a la predicció d'una sèrie de dades; llavors, s'ha de configurar la xarxa per a que a la sortida es tingui una capa de regressió.



Il·lustració 4-10: App per a dissenyar xarxes neuronals amb diagrames de blocs

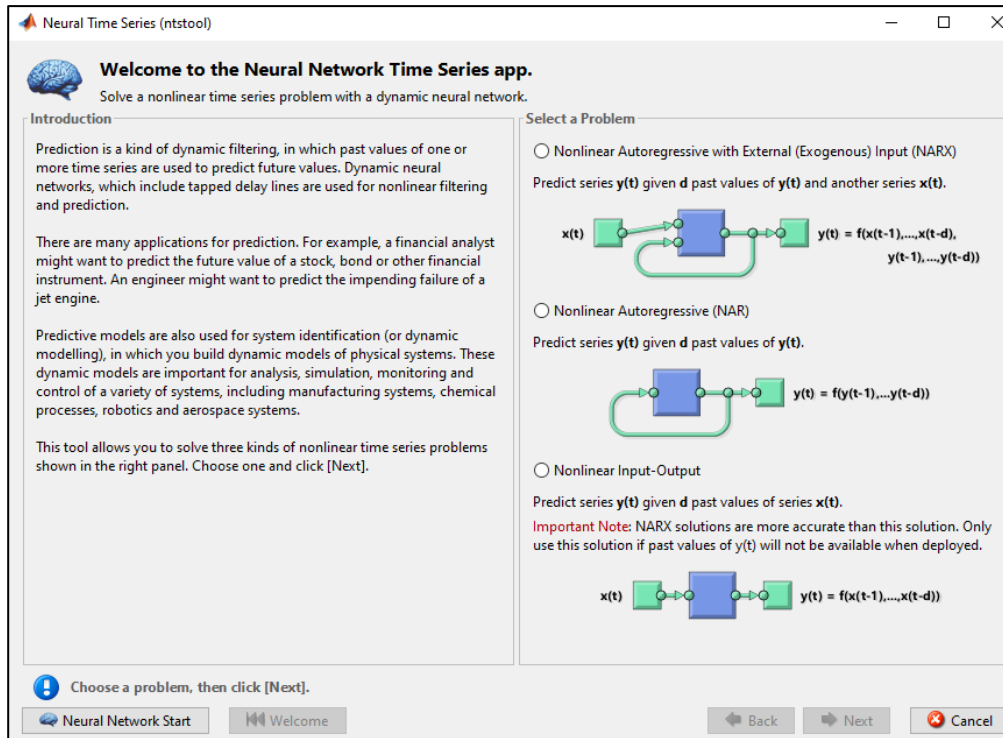
A més a més, es pot accedir als codis d'aquestes apps; per tant, un cop feta una programació més intuïtiva, es pot treballar amb el seu codi, tal i com es mostra a la *Il·lustració 4-11*.



Il·lustració 4-11: Opció de generar codi amb l'app del disseny de xarxes neuronals

En relació a la informació donada al llarg d'aquest document, les dades que alimenten la xarxa són una seqüència temporal. Aleshores, hi ha un altre programa dins de *Matlab* que pot tenir importància: *Neural Time Series*. De la mateixa manera que a la darrera app, aquesta també pot

imprimir el seu codi.



Il·lustració 4-12: Pàgina introductòria de l'app Neural Network Time Series

La particularitat d'aquestes eines és la que segueix: totes tenen funcions dins de *Matlab* per a dissenyar i entrenar les xarxes, però el test va ser programat pel autor d'aquest treball. A més a més, altre particularitat de aquestes dues apps és que la inicialització dels paràmetres (bias, pesos...) es fa amb valors de 0; llavors, ja es té una possibilitat menys d'aconseguir la xarxa neuronal òptima.

A les prediccions de la generació, es van utilitzar xarxes neuronals artificials amb capes ocultes **LSTM**. Hi ha un motiu essencial per a la selecció d'aquesta xarxa: permet treballar amb una quantitat alta de dades [8], pel que es pot fer l'entrenament de dades plantejat al darrer apartat. Aquí, les paràmetres variats van ser:

- Mètode d'entrenament.
- Èpoques d'entrenament.

- Número de capes ocultes.
- Número de neurones dins de cadascuna de les capes ocultes.
- Tassa d'aprenentatge.

Per a les prediccions de consum, es va utilitzar la xarxa **Nonlinear Input-Output**, de l'app de *Neural Network Time Series*.

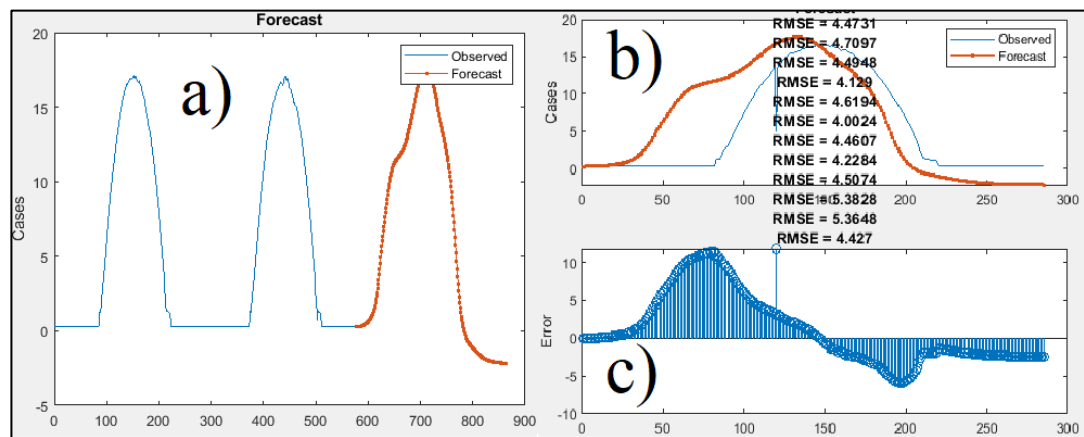
5. Resultats

5.1. Predicció generació australiana

Tal i com s'ha dit al capítol 4, per a la generació dels dades del projecte es van programar xarxes neuronals artificials LSTM.

Per aquesta part, només es mostrarà una de les xarxes dissenyats, ja que, com es sap, aquesta part no és més que introductòria. Així doncs, la xarxa de la qual, els seus resultats apareixen a la *Il·lustració 5-1*, té aquestes propietats:

- Algoritme d'entrenament: Adam.
- 2 capes ocultes, tipus LSTM, on la primera té 82 neurones i la segona, 90 (resultat de prova i error).
- Tassa d'aprenentatge: 0,0025 (referent a la problemàtica explicada al capítol 3).
- Èpoques d'iteració: 100 (es veu com l'error convergeixi abans de completar totes les iteracions).



Il·lustració 5-1: : Resultat per a la predicció del primer dia (test), per a la base de dades australiana. a) mostra la radiació predita; b) fa la comparació entre la radiació predita i el seu valor real; i c) mostra l'error corresponent a la diferència de la predicció i el valor real.

A la figura a) es mostra, en color taronja, la predicció feta pel tercer dia. A la b), fa la comparació entre aquesta predicció i el seu valor real, mostrant, a la c), el error quadràtic d'aquests dos valors. El temps d'entrenament va ser menor a 2 minuts.

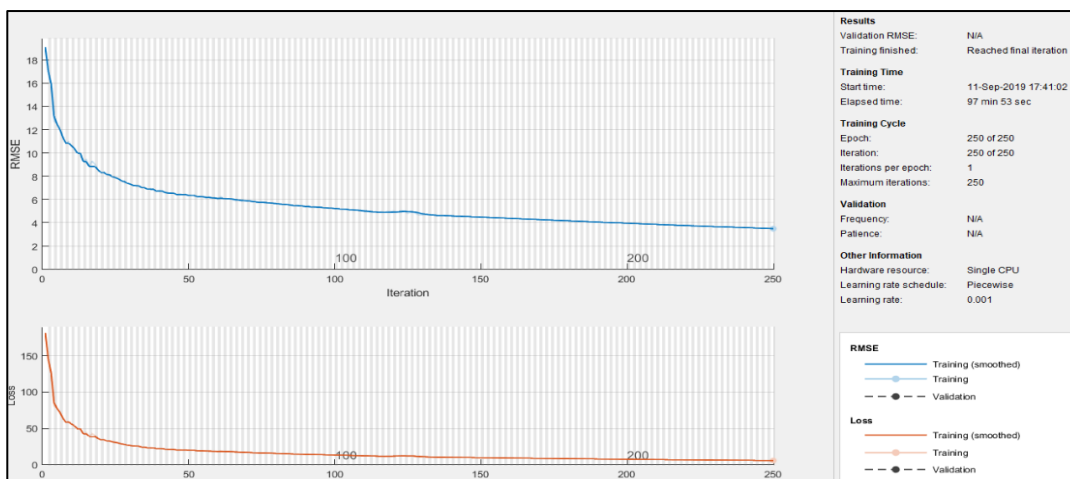
5.2. Predicció generació dades del projecte

A continuació, es mostren els diferents resultats per una simulació feta per a unes mateixes xarxes neuronals. A diferència del cas anterior, aquí el vull fer un anàlisi més detallat; en primer lloc, cal dir que els resultats obtinguts en totes les simulacions no van ser mostrats a la memòria. Només apareixen les xarxes neuronals que donen resultat acceptable.

Llavors, atès que ara es treballa amb més dades, el període d'entrenament és més llarg, i també ho són les capes i el número de neurones a les capes. Dins de les proves fetes, una xarxa que té una resposta acceptable amb els 3 mètodes d'entrenament va ser aquesta amb aquests paràmetres:

- 3 capes ocultes, tipus LSTM, on la primera té 400 neurones, la segona 300 i la tercera 300.
- Tassa d'aprenentatge: 0,005.
- Èpoques d'iteració: 250.

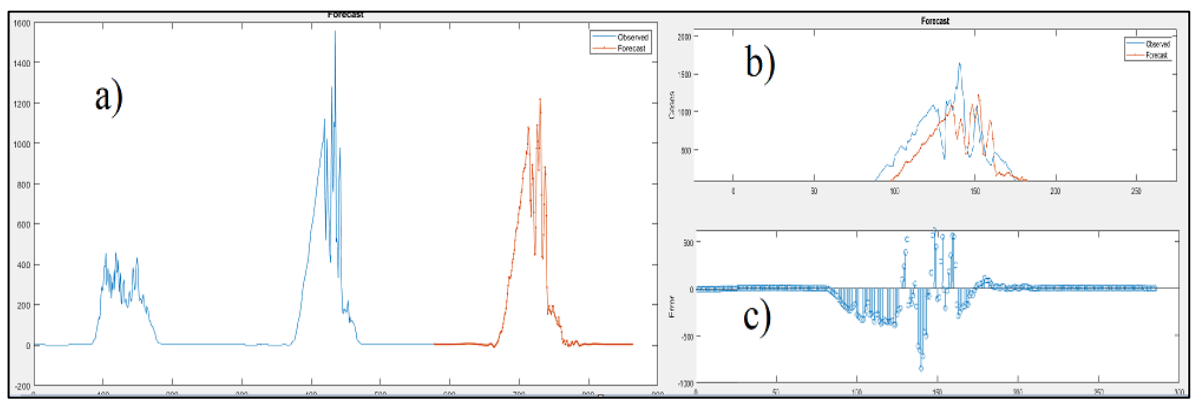
A la *Il·lustració 5-2* es pot veure el decrement de l'error durant les èpoques d'entrenament, utilitzant l'algoritme *Adam*. Es pot veure que encara no convergeixi del tot, fixant les èpoques màximes en 250; el temps de l'entrenament va ser de 97 minuts.



Il·lustració 5-2: Procés de l'entrenament de la xarxa, amb l'algoritme Adam

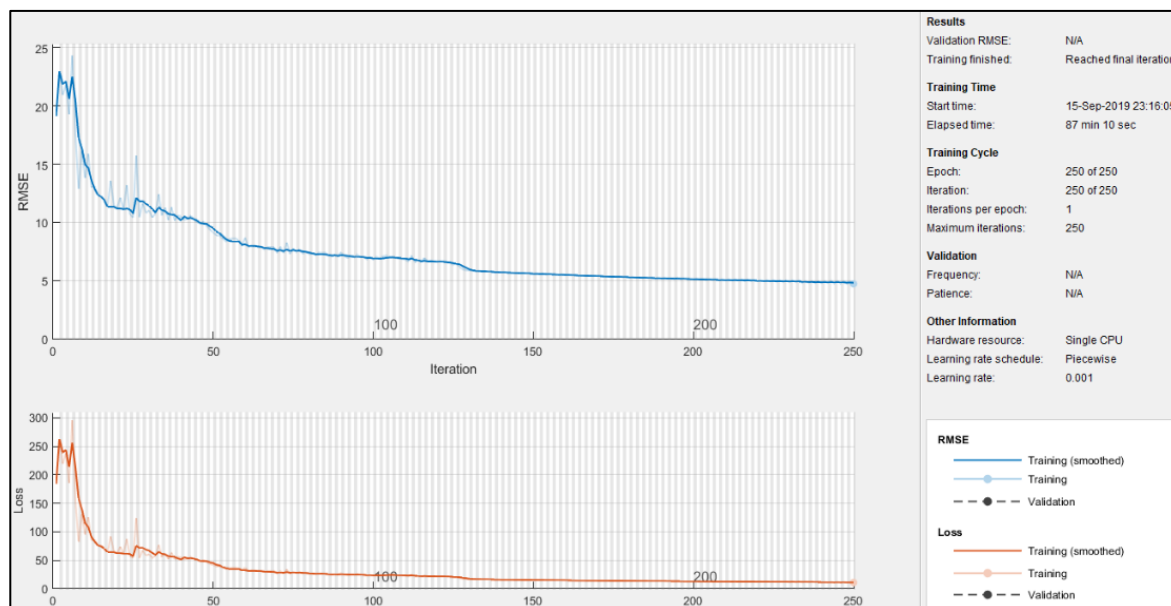
A continuació, a la *Il·lustració 5-3* es mostra la primera predicció feta (és a dir, el primer i segon dia són l'entrenament, i el tercer, és la predicció o test). A la part *a)* estan dibuixades, en blau, les dades del primer i segon dia, així com la predicció, que correspon al tercer dia, té el color taronja. Es pot veure com la xarxa segueix més la senyal del darrer dia que del primer, al moment de fer la predicció.

Després, a la part *b)*, es fa la comparació entre el valor real de la radiació al tercer dia (blau) i la predicció (taronja), mentre que a la *c)* es veu, per a cadascun dels intervals, l'error a la predicció.



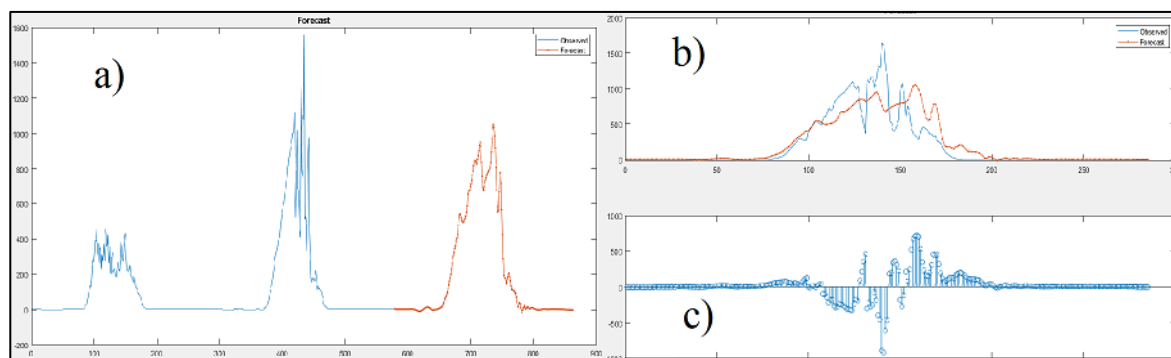
Il·lustració 5-3: Resultat per a la predicció del primer dia (test), utilitzant l'algoritme Adam.. a) mostra la radiació predita; b) fa la comparació entre la radiació predita i el seu valor real; i c) mostra l'error corresponent a la diferència de la predicció i el valor real.

A la *Il·lustració 5-4*, en canvi, es mostra la tendència de l'error durant l'entrenament amb l'algoritme Rmsprop. L'error s'estabilitza abans que l'Adam, però ho fa tenint més error.



Il·lustració 5-4: Procés de l'entrenament de la xarxa, amb l'algoritme Rmsprop

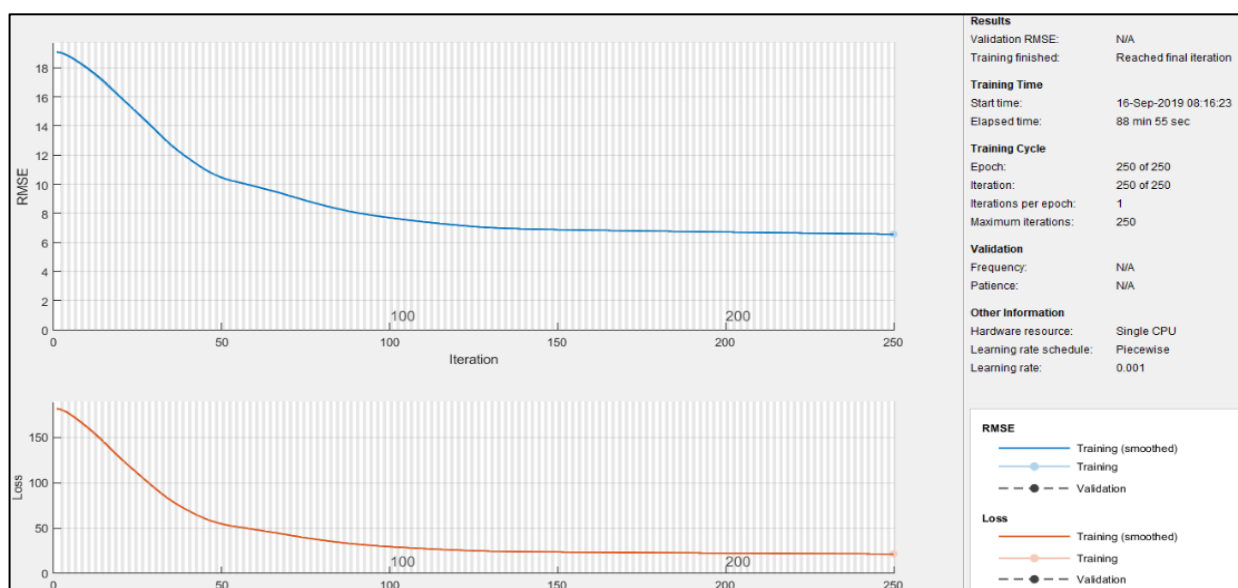
El seu temps de computació, en canvi, és menor. A la Il·lustració 5-5 es veu, també, la primera predicció de totes. En aquest cas, la predicció feta agafa valors intermedis dels dos dies d'entrenament, tal i com apareix a la par a). Un cop més, a la imatge b) es pot recollir la comparació entre la predicció i el seu valor de veritat, mostrant en c) els valors de l'error, per aquesta predicció.



Il·lustració 5-5: Resultat per a la predicció del primer dia (test), utilitzant l'algoritme Rmsprop. a) mostra la radiació predita; b) fa la comparació entre la radiació predita i el seu valor real; i c) mostra l'error corresponent a la diferència de la predicció i el valor real.

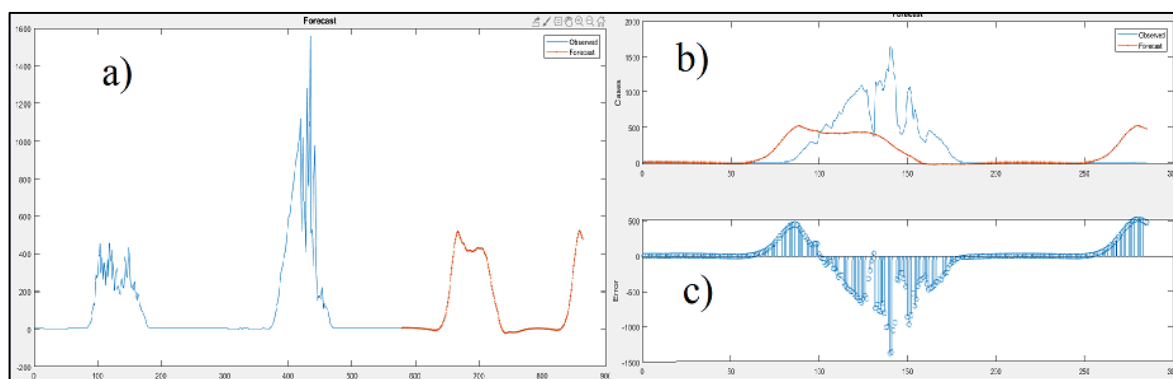
Ara doncs, és el torn del disseny fet a la xarxa amb l'algoritme SGDM. La Il·lustració 5-6 diu

que aquest mètode, encara que convergeix al seu estat estacionari massa ràpid, té un error més gran.



Il·lustració 5-6: Procés de l'entrenament de la xarxa, amb l'algoritme SGDM.

Finalment, es fa l'anàlisi de la predicció feta. Amb aquest algoritme, se li dona més importància al primer dia d'entrenament que al segon (Il·lustració 5-7, a)). A més a més, la predicció feta, aquesta vegada, és la menys encertada.



Il·lustració 5-7: Resultat per a la predicció del primer dia (test), utilitzant l'algoritme SGDM. a) mostra la radiació predita; b) fa la comparació entre la radiació predita i el seu valor real; i c) mostra l'error corresponent a la diferència de la predicció i el valor real.

Aquests resultats mostrats són una petita part dels experiments fets. Ara bé, s'ha fet el recompte del error global per a cadascun dels algoritmes, per a aquesta xarxa, i el resultat és el que segueix, a la Taula 5-1:

Algoritme	Adam	Rmsprop	SGDM
Error global	6,6401E+07	7,3454E+07	8,1899E+07

Taula 5-1: Errors globals de la mateixa xarxa canviant només l'algoritme d'entrenament

És necessari explicar quin tipus d'error s'ha calculat: es tracta de la suma dels errors mitjans de cadascun dels tests. Això es fa perquè, a les prediccions, ocorren errors durant les hores de sol; generalment, a la nit prediuen la absència de radiació. A més a més, l'error és diferent al llarg de l'any.

Aquestes valors són només per a aquesta xarxa neuronal. Però, és cert que, fent diferents xarxes, el comportament d'aquestes, s'assemblen a les tendències ofertes en aquest punt segons els seus entrenaments.

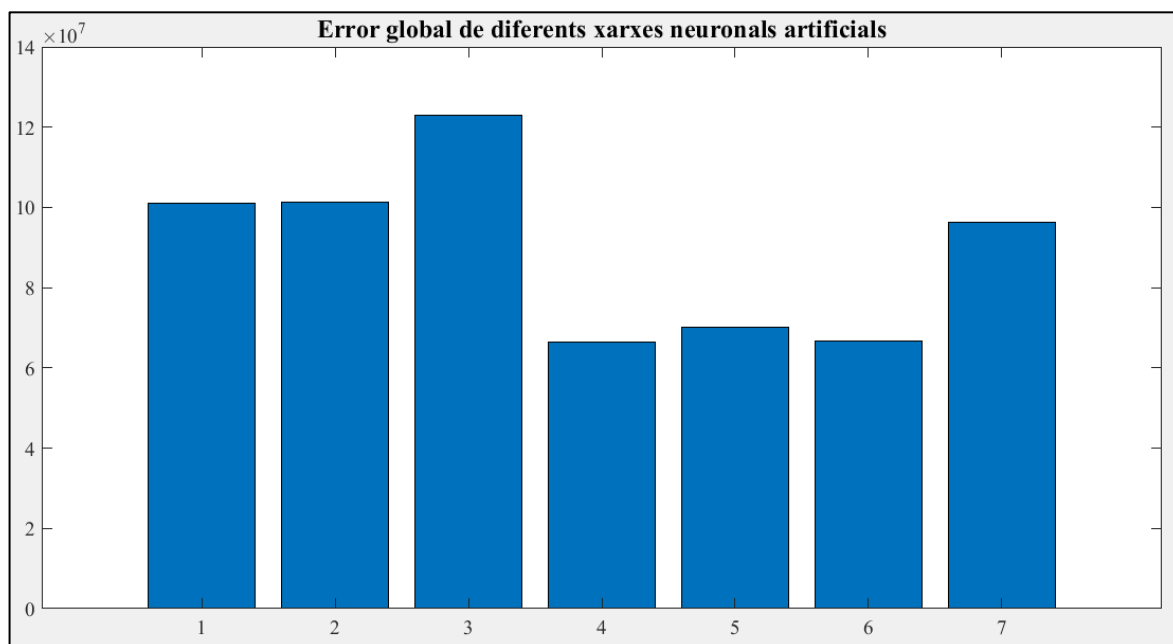
Llavors, a partir d'ara el paràmetre que es fixa és l'algoritme d'entrenament: **Adam**. Nombroses arquitectures van fer fetes amb xarxes neuronals artificials LSTM. A continuació, a la *Taula 5-2*, es mostren els resultats de l'error obtinguts per a la composició de les xarxes amb un millor resultat obtingut.

Nombre de xarxa	Nombre de capes ocultes	Nombre de neurones en cadascuna de les capes	Error global
1	3	400-300-300	1,0098E+08
2	3	500-350-300	1,0128E+08
3	3	430-320-310	1,2285E+08
4	3	400-300-300	6,6401E+07
5	3	500-350-300	7,0108E+07
6	4	800-750-700-650	6,6583E+07
7	6	700-800-900-1000-900-750	9,6212E+07

Taula 5-2: Composició de les xarxes analitzades a per l'algoritme d'entrenament Adam.

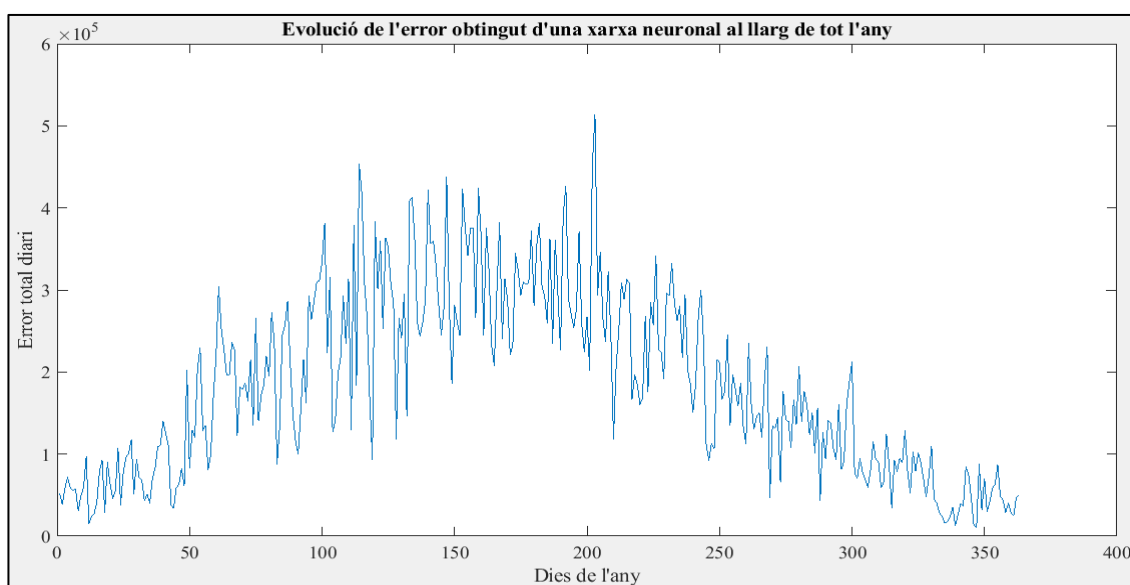
A més, a manera de mostrar d'una manera més gràfica l'error global d'aquestes xarxes neuronals artificials, es presenta la *Il·lustració 5-8*. A manera de comentari, la xarxa que millor

resultat ofereix, segons el seu error, és la de 3 capes ocultes, amb 400 neurones a la primera capa, 300 a la segona i 300 a la tercera.



Il·lustració 5-8: Errors globals de les xarxes neuronals artificials LSTM dissenyades amb l'algorisme Adam

Per concloure amb aquest apartat, a la *Il·lustració 5-9* es vull ensenyar com evoluciona l'error mesurat al llarg de l'any sencer. Agafant l'error diari total de la xarxa amb menor error obtingut, es construeix la evolució anual del seu error.



Il·lustració 5-9: Variació de l'error diari al llarg de tot l'any, per a la xarxa neuronal artificial amb menor error global obtinguda

Com es pot sentir, aquest és màxim als dies centrals de l'any, és a dir, als corresponents a l'estiu; però com l'error augmenta de forma proporcional com ho fa la radiació a l'estiu, es considera que la mitjana de l'error és la mateixa en tot l'any; ho mateix podria passar amb la seva dispersió.

5.3. Predicció consum dades del projecte

Per iniciar amb aquest apartat, es va començar amb una xarxa “fictícia”. És a dir, amb l'objectiu de veure si existeix correlació entre les prediccions fetes i el consum de la microxarxa, la primera xarxa que es va dissenyar va ser amb dades de generació reals; com si es tractessin de resultats de les prediccions fetes a l'apartat anterior amb una exactitud del 100%.

Llavors, la xarxa proposada va ser *Nonlinear Input-Output*, de l'eina *Neural Net Time Series*. Tal i com s'ha dit abans, es vull lligar el consum amb les prediccions de la generació; aleshores, no es pot utilitzar la mateixa estratègia de la predicció de la generació amb xarxes LSTM. Una gran falta d'aquest tipus de xarxa és que no pot funcionar bé amb una sèrie de dades tan llarga com la del projecte.

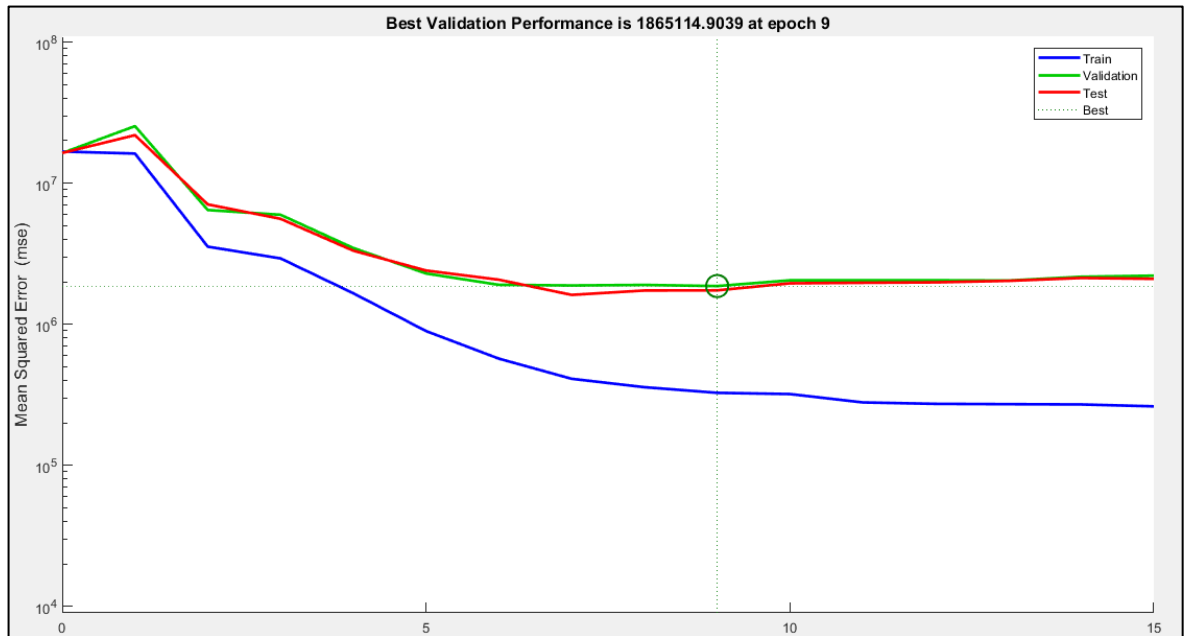
Llavors, hi ha dues opcions:

- **Opció 1:** Fer un filtratge dels dades d'entrada i sortida de la xarxa, com poden ser eliminar les hores de la nit i treballar amb els valors horaris medis de la resta de les dades.
- **Opció 2:** Deixant els dades de la nit i utilitzant les dades 5-minutals, treballar amb una xarxa neuronal amb paràmetres bàsics, per no sobrecarregar la capacitat computacional de *Matlab*.

La decisió presa va ser la segona; abans, al capítol 4, es va decidir no fer cap filtratge, per a treballar amb les dades més propers a la realitat. La xarxa programada, va tenir les següents paràmetres:

- 10 neurones a la única capa oculta que hi ha.
- Un *delay* de 2 unitats de temps.

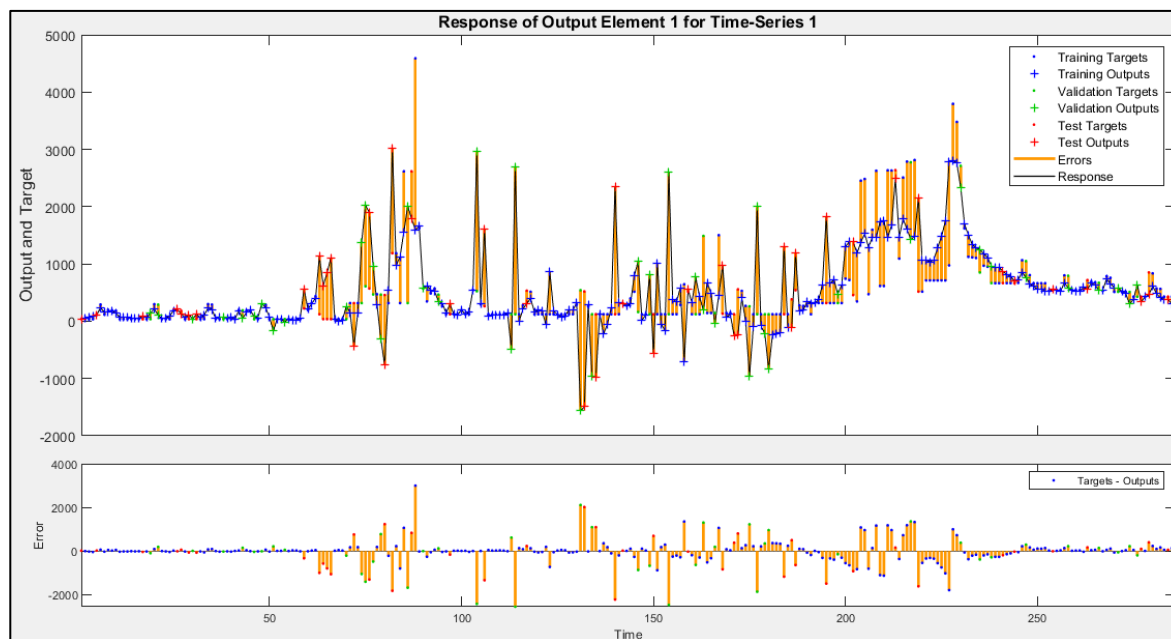
Un cop executada la simulació, els resultats van ser:



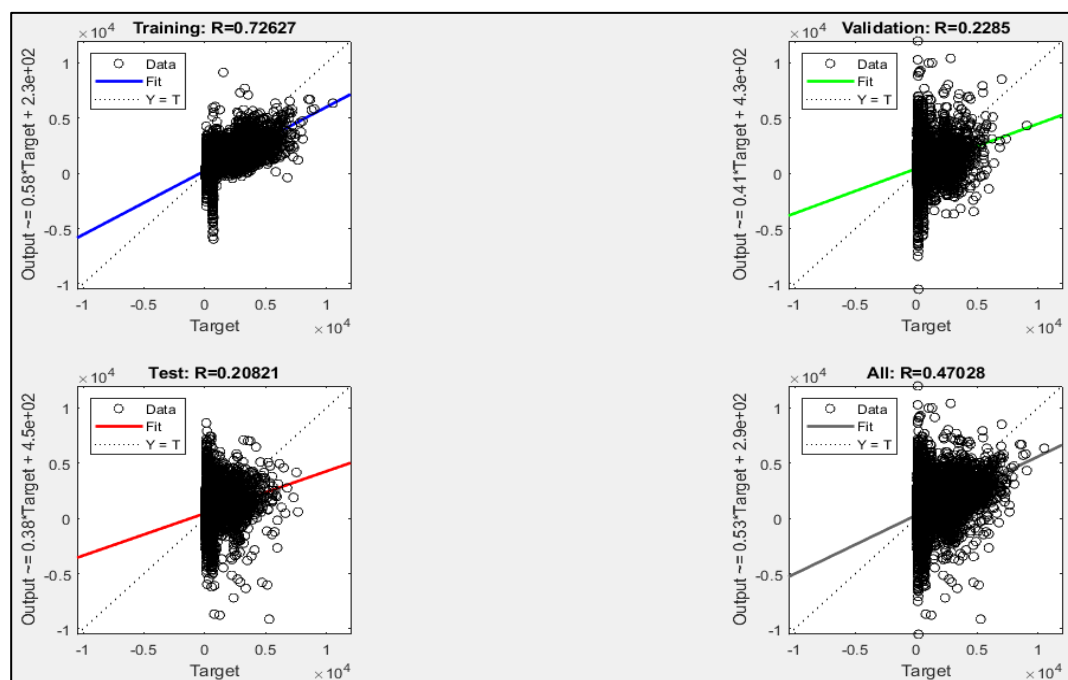
Il·lustració 5-10: Selecció de la millor actuació del test segons el seu error.

A la *Il·lustració 5-7* es veu com són els paràmetres dissenyats a la 9^a època les que menys error agafen en el moment de la validació. Això vull dir que amb aquesta eina, *Matlab* pot recordar en cadascuna de les seves iteracions els valors dels paràmetres de la xarxa.

A continuació, a la *Il·lustració 5-8*, apareix el la representació gràfica de la sortida de la primera predicció. Un cop més, l'error més gros està a les hores que hi ha radiació; els intervals de la nit, els segueix bé, gairebé sense error.



Il·lustració 5-11: Resposta del primer element de la xarxa neuronal, amb les seves mesures de l'error.



Il·lustració 5-12: Per a la sortida del primer element de la xarxa, error de l'entrenament (dalt, esquerra), error de validació (dalt, dreta), error del test (baix, esquerra) i el error global (baix, dreta).

Finalment, a la Il·lustració 5-9 es mostren els errors quadràtics de l'entrenament, la validació,

el test i el global. Com aspecte fonamental per al dolent funcionament d'aquesta xarxa, en aquesta il·lustració es quantifiquen les prediccions negatives que surten; això, tractant-se de la generació fotovoltaica, que deriva directament del fenomen físic radiació fotovoltaica, és un fet impossible.

6. Conclusions i mires cap al futur

6.1. Conclusions sobre les prediccions fetes de la generació

La primera conclusió que es fa és que les eines utilitzades amb el software *Matlab*, encara que no és el més potent, serveixen per a fer prediccions de series temporals amb caràcter no lineal. A més a més, per a les prediccions de generació fotovoltaica tenint una quantitat elevada de dades, amb *Matlab* el millor tipus de xarxa neuronal artificial és la LSTM.

En aquesta part del projecte es van analitzar dades de tipus no lineal, amb caràcter temporal. Per a aquests tipus de dades, dins de les xarxes LSTM, les paràmetres com la tasa d'aprenentatge i les èpoques màximes per a les iteracions, es poden aconseguir gracies a bibliografia o treballs anteriors. Després, a l'elecció de les mètodes d'entrenament, és cert que el que millor funciona, per aquest tipus de dades, és l'algoritme Adam. Tot i així, l'elecció d'un altre algoritme d'entrenament no farà que la xarxa funcioni malament, si el rest dels paràmetres estan bé fixats.

Llavors, els paràmetres fonamentals al disseny d'aquesta xarxa neuronal va ser, en primer lloc, el nombre de capes ocultes de la xarxa, i en segon lloc, el nombre de neurones dins de cadascuna de aquestes capes. Aquesta part va ser completament iterativa; els resultats obtinguts al llarg de tot el període de disseny de xarxes no van ser mostrats a l'apartat 5. A més a més, per als dades d'Austràlia, les quantitat de capes i neurones que hi ha en cadascuna d'elles va ser menor per una obtenció de predicció acceptable.

També cal dir que hi ha cassos en que la variació del 2% de les neurones dins d'una capa oculta d'una xarxa que tenia un comportament acceptable, va provocar que la nova xarxa no predir cap valor.

En conclusió, el disseny de xarxes neuronals LSTM depenen, en un gran valor, de la naturalesa i la mida dels dades d'entrada, ja que és ho que fa que aquest problema sigui únic.

6.2. Conclusions sobre les prediccions fetes del consum

Mitjançant les eines que ofereix *Matlab*, i la determinació de l'autor d'aquest projecte de voler treballar amb dades sense filtracions ni eliminacions d'hores de nit, no es va poder fer un disseny acceptable per a lligar la predicció feta de la generació amb les dades de consum.

Fins i tot, es va procedir a realitzar les prediccions del consum amb uns dades de generació predita “ideals”, com si fossin obtingudes amb una exactitud del 100%, però la impossibilitat de treballar amb una xarxa de major capacitat no deixa obtenir una resposta clara a aquesta problemàtica.

6.3. Mires cap al futur

En primer lloc, seria un bon experiment tractar de fer els mateixos dissenys per a les xarxes d'aquest treball, però incloent més dades d'entrada: dades dels històrics de radiació i consum de 2 o 3 anys, per exemple. Altres dades extres podrien ser afegits: la humidesa, la nebulositat i la temperatura de la zona. Aquestes paràmetres acompanyen prou bé a la predicció de generació fotovoltaica [9].

En segon lloc, caldria provar amb altres softwares de disseny de xarxes neuronals artificials. Encara que s'ha utilitzat prou bé *Matlab*, és important fer dissenys amb els mateixes dades en softwares diferents.

En tercer lloc, tot per aconseguir el millor mètode, altra possibilitat a explorar seria la utilització d'altres mètodes de regressió, com SVM [8]. No es tracten de xarxes neuronals artificials, però resulta de gran interès explorar en diferents tecnologies una possible solució a la predicció de fenòmens naturals com la radiació solar.

Finalment, i tornant a les xarxes neuronals artificials, cal insistir en la singularitat d'aquesta tecnologia. Les solucions presentades al treball de fi de màster que funcionen, no són les

millores; aleshores, hi ha un gran camí per fer, on les possibilitats acaricien l'infinit.

Bibliografía

- [1] CogniFit, “Las Neuronas: Razones por las que nuestras neuronas pueden vivir mejor y por más tiempo.” [Online]. Available: <https://www.cognifit.com/es/neuronas>.
- [2] A. M. Turing, “Computing Machinery and Intelligence,” *Mind*, vol. 49, pp. 433–460, 1950.
- [3] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [6] DKA Solar Centre, “DKASC, Alice Spring.” [Online]. Available: <http://dkasolarcentre.com.au/locations/alice-springs/graphs?sources=71>.
- [7] A. Alzahrani, S. Pourya, C. Dagli, and M. Ferdowsi, “Solar Irradiance Forecasting Using Deep Neural Networks,” *ScienceDirect*, vol. 114, pp. 304–313, 2017.
- [8] R. Bonetto and M. Rossi, “Machine Learning Approaches to Energy Consumption Forecasting in Households,” Padova, 2017.
- [9] A. Kuzmiakova, G. Colas, and A. McKeehan, “Short-term Memory Solar Energy Forecasting at University of Illinois,” *Univ. Illinois*, pp. 1–6, 2017.

